Learning Graphs from Data

Gonzalo Mateos

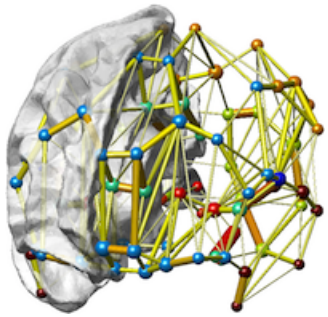
Dept. of ECE and Goergen Institute for Data Science
University of Rochester
gmateosb@ece.rochester.edu
http://www.ece.rochester.edu/~gmateosb/

Acknowledgment: Santiago Segarra

Facultad de Ingeniería, UdelaR
Montevideo, Uruguay
February 5, 2020

What is this lecture about?

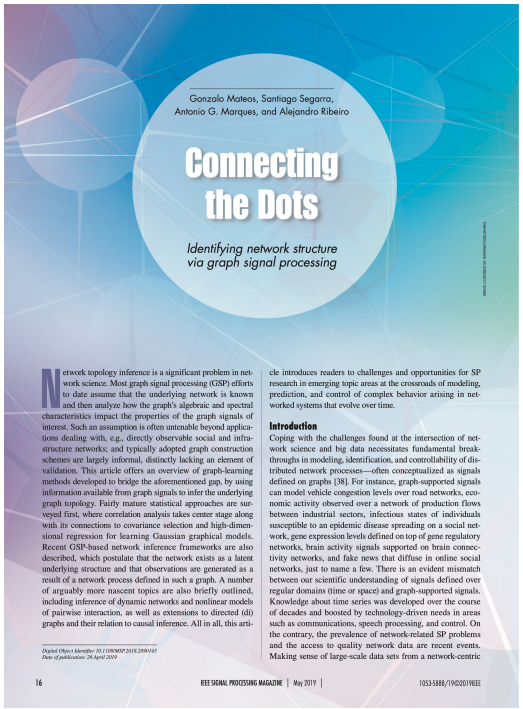
- Learning graphs from nodal observations
- Ex: Central to network neuroscience
  $\Rightarrow$ Functional network from fMRI signals

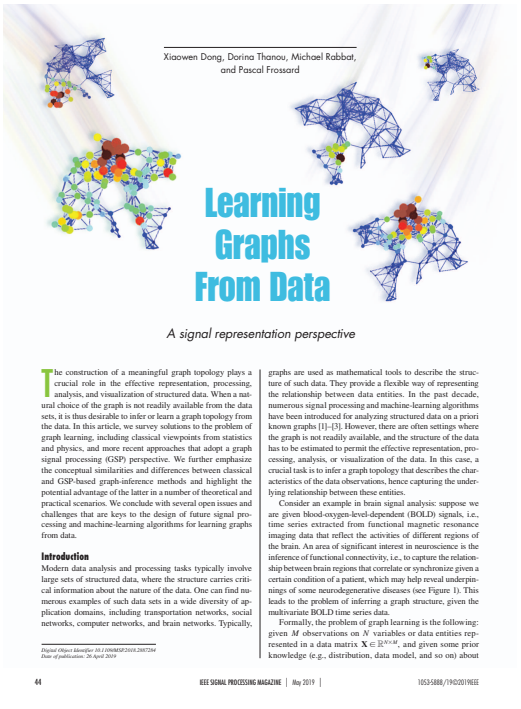


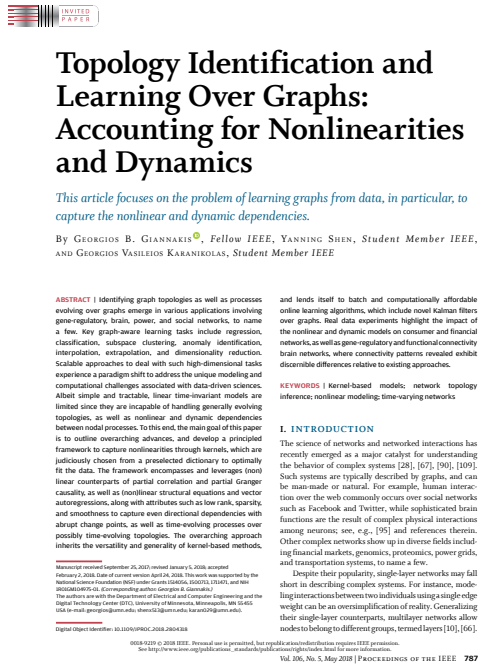
- Most GSP works: how known graph G affects signals and filters
  - Feasible for e.g., physical networks
  - Links are tangible and directly observable
- Still, acquisition of updated topology information is challenging
  $\Rightarrow$ Sheer size, reconfiguration, privacy and security
- Here, reverse path: how to use GSP to infer the graph topology?
- **Goal:** recover a latent network, or, a graph-based data representation

Connecting the dots

- Recent tutorials on learning graphs from data
  - IEEE Signal Processing Magazine and Proceedings of the IEEE







- IEEE Trans. on Signal and Information Processing over Networks
  - Special issue on Network Topology Inference (Jan. 2020)

Network topology inference

Statistical methods for network topology inference

Learning graphs from observations of smooth signals

Identifying the structure of network diffusion processes

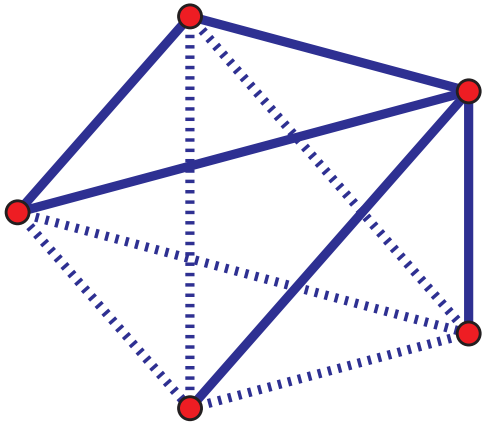
Network topology inference problems

- **Q:** If G (or a portion thereof) is unobserved, can we infer it from data?
- Formulate as a statistical inference task, i.e. given
  - Signal measurements x_i at some or all vertices $i \in \mathcal{V}$
  - Indicators A_{ij} of edge status for some vertex pairs $\{i,j\} \in \mathcal{V}^{(2)}_{obs}$
  - A collection $\mathcal{G}$ of candidate graphs G

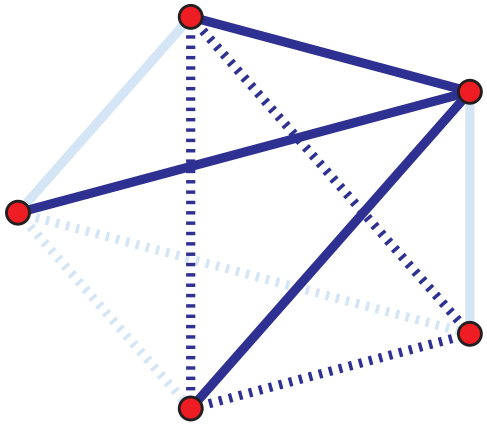
  Goal: infer the topology of the network graph $G(\mathcal{V},\mathcal{E})$
- Bring to bear existing statistical concepts and tools

  $\Rightarrow$ Study identifiability, consistency, robustness, complexity
- Three canonical network topology inference problems [Kolaczyk'09]
  (i) Link prediction
  (ii) Association network inference $\leftarrow$ Focus of this lecture
  (iii) Tomographic network topology inference

Link prediction



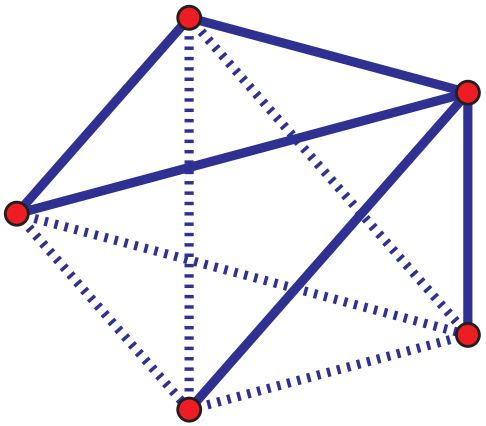
Original graph



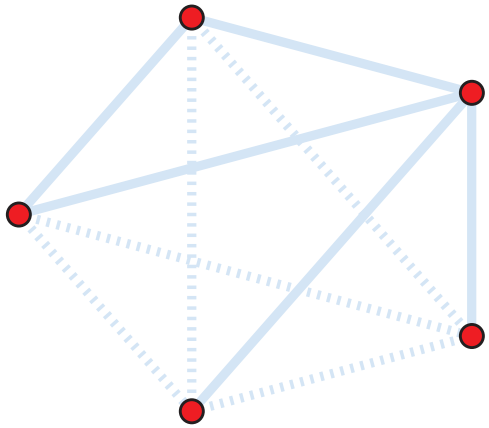
Link prediction

- Suppose we observe the graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top$; and
- Edge status is only observed for some subset of pairs $\mathcal{V}^{(2)}_{obs} \subset \mathcal{V}^{(2)}$
- **Goal:** predict edge status for all other pairs, i.e., $\mathcal{V}^{(2)}_{miss} = \mathcal{V}^{(2)} \setminus \mathcal{V}^{(2)}_{obs}$

Association network inference



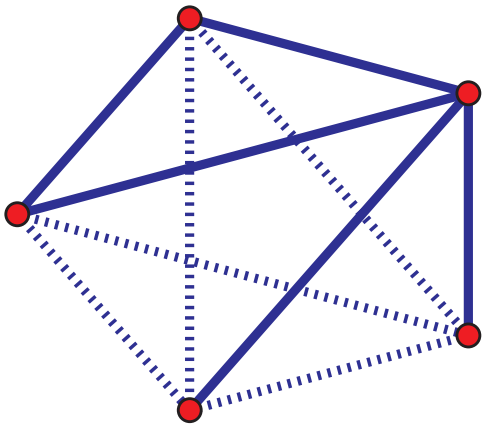
Original graph



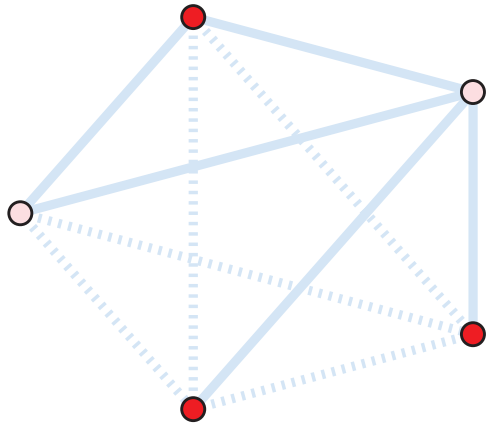
Association network inference

- Suppose we only observe the graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top$; and
- Assume (i, j) defined by nontrivial 'level of association' among x_i, x_j
- **Goal:** predict edge status for all vertex pairs $\mathcal{V}^{(2)}$

Tomographic network topology inference



Original graph



Tomographic inference

- Suppose we only observe x_i for vertices $i \subset \mathcal{V}$ in the 'perimeter' of G
- **Goal:** predict edge and vertex status in the 'interior' of G

Association network inference

- Given a collection of N elements represented as vertices $v \in \mathcal{V}$
  - Graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$ of observed vertex attributes
- User-defined similarity $\texttt{sim}(i,j) = f(x_i, x_j)$ specifies edges $(i,j) \in \mathcal{E}$
  - **Q:** What if `sim` values themselves (i.e., edge status) not observable?

Association network inference

Infer non-trivial `sim` values from i.i.d. observations $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$

- Various choices to be made, hence multiple possible approaches
  - Choice of `sim`: correlation, partial correlation, mutual information
  - Choice of inference: hypothesis testing, regression, ad hoc
  - Choice of parameters: testing thresholds, tuning regularization

Correlation networks

- Pearson product-moment correlation as `sim` between vertex pairs

$$\texttt{sim}(i,j) := \rho_{ij} = \frac{\text{cov}[x_i, x_j]}{\sqrt{\text{var}\,[x_i]\,\text{var}\,[x_j]}}, \quad i,j \in \mathcal{V}$$

- **Def:** the correlation network graph $G(\mathcal{V}, \mathcal{E})$ has edge set

$$\mathcal{E} = \left\{(i,j) \in \mathcal{V}^{(2)} : \rho_{ij} \neq 0\right\}$$

  - Association network inference $\Leftrightarrow$ Inference of non-zero correlations

- Inference of $\mathcal{E}$ typically approached as a testing problem

$$H_0 : \rho_{ij} = 0 \quad \text{versus} \quad H_1 : \rho_{ij} \neq 0$$

Test statistics

- Common choice of test statistic are empirical correlations

$$\hat{\rho}_{ij} = \frac{\hat{\sigma}_{ij}}{\sqrt{\hat{\sigma}_{ii}\hat{\sigma}_{jj}}}, \quad \text{where } \hat{\mathbf{\Sigma}} = [\hat{\sigma}_{ij}] = \frac{1}{P-1}\sum_{p=1}^{P} \mathbf{x}_p \mathbf{x}_p^\top$$

- Convenient alternative statistic is Fisher's transformation

$$\hat{z}_{ij} = \frac{1}{2}\log\left(\frac{1+\hat{\rho}_{ij}}{1-\hat{\rho}_{ij}}\right), \quad i,j \in \mathcal{V}$$

$\Rightarrow$ Under H_0, $\hat{z}_{ij} \sim \mathcal{N}(0, \frac{1}{P-3})$ $\Rightarrow$ Simple to assess significance

- Reject H_0 at significance level α, i.e., assign edge (i,j) if $|\hat{z}_{ij}| > \frac{z_{\alpha/2}}{\sqrt{P-3}}$

Error rate control: $\mathrm{P}_{H_0}(\text{false edge}) = \mathrm{P}_{H_0}\left(|\hat{z}_{ij}| > \frac{z_{\alpha/2}}{\sqrt{P-3}}\right) = \alpha$

Networks and multiple testing

- Interesting testing challenges emerge with large-scale networks
  - $\Rightarrow$ Suppose we test all $\binom{N}{2}$ vertex pairs, each at level α
- Even if the true G is the empty graph, i.e., $\mathcal{E} = \emptyset$
  - $\Rightarrow$ We expect to declare $\binom{N}{2}\alpha$ spurious edges just by chance!
  - $\Rightarrow$ For a large graph, this number can be considerable
- Ex: For G of order $N = 100$ and individual tests at level $\alpha = 0.05$
  - $\Rightarrow$ Expected number of spurious edges is $4950 \times 0.05 \approx 250$
- This predicament known as the multiple testing problem in statistics

Correction for multiple testing

- **Idea:** Control errors at the level of collection of tests, not individually
- False discovery rate (FDR) control, i.e., for given level γ ensure

$$\text{FDR} = \mathbb{E}\left[\frac{R_{false}}{R} \,\middle|\, R > 0\right] \text{P}\left[R > 0\right] \leq \gamma$$

  - R is the total number of edges detected; and
  - R_{false} is the number of false edges detected

- Method of FDR control at level γ [Benjamini-Hochberg'94]

  Step 1: Sort p-values for all $\bar{N} := \binom{N}{2}$ tests, yields $p_{(1)} \leq \ldots \leq p_{(\bar{N})}$

  Step 2: Reject H_0, i.e., declare all those edges for which

$$p_{(k)} \leq \left(\frac{k}{\bar{N}}\right)\gamma$$

Partial correlations

- Use correlations carefully: 'correlation does not imply causation'
  - Vertices $i, j \in \mathcal{V}$ may have high ρ_{ij} because they influence each other
- But ρ_{ij} could be high if both i, j influenced by a third vertex $k \in \mathcal{V}$

  $\Rightarrow$ Correlation networks may declare edges due to confounders
- Partial correlations better capture direct influence among vertices
  - For $i, j \in \mathcal{V}$ consider latent vertices $S_m = \{k_1, \ldots, k_m\} \subset \mathcal{V} \setminus \{i, j\}$
- Partial correlation of x_i and x_j, adjusting for $\mathbf{x}_{S_m} = [x_{k_1}, \ldots, x_{k_m}]^\top$ is

$$\rho_{ij|S_m} = \frac{\operatorname{cov}[x_i, x_j \,|\, \mathbf{x}_{S_m}]}{\sqrt{\operatorname{var}\left[x_i \,|\, \mathbf{x}_{S_m}\right] \operatorname{var}\left[x_j \,|\, \mathbf{x}_{S_m}\right]}}, \quad i, j \in \mathcal{V}$$

- **Q:** How do we obtain these partial correlations?

Computing partial correlations

- Given $\mathbf{x}_{S_m} = [x_{k_1}, \ldots, x_{k_m}]^\top$, the partial correlation of x_i and x_j is

$$\rho_{ij|S_m} = \frac{\text{cov}[x_i, x_j \,|\, \mathbf{x}_{S_m}]}{\sqrt{\text{var}\left[x_i \,|\, \mathbf{x}_{S_m}\right] \text{var}\left[x_j \,|\, \mathbf{x}_{S_m}\right]}} = \frac{\sigma_{ij|S_m}}{\sqrt{\sigma_{ii|S_m}\sigma_{jj|S_m}}}$$

- Here $\sigma_{ii|S_m}, \sigma_{jj|S_m}$ and $\sigma_{ij|S_m}$ are diagonal and off-diagonal elements of

$$\mathbf{\Sigma}_{11|2} := \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21} \in \mathbb{R}^{2\times 2}$$

- Matrices $\mathbf{\Sigma}_{11}$, $\mathbf{\Sigma}_{22}$ and $\mathbf{\Sigma}_{21} = \mathbf{\Sigma}_{12}^\top$ are blocks of the covariance matrix

$$\text{cov}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{pmatrix}, \quad \text{where } \mathbf{w}_1 := [x_i, x_j]^\top \text{ and } \mathbf{w}_2 := \mathbf{x}_{S_m}$$

Partial correlation networks

- Various ways to use partial correlations to define edges in G

  Ex: x_i, x_j correlated regardless of what m vertices we condition upon

$$\mathcal{E} = \left\{ (i,j) \in \mathcal{V}^{(2)} : \rho_{ij|S_m} \neq 0, \text{ for all } S_m \in \mathcal{V}^{(m)}_{\setminus\{i,j\}} \right\}$$

- Inference of potential edge (i,j) as a testing problem

$$H_0 : \rho_{ij|S_m} = 0 \text{ for some } S_m \in \mathcal{V}^{(m)}_{\setminus\{i,j\}}$$

$$H_1 : \rho_{ij|S_m} \neq 0 \text{ for all } S_m \in \mathcal{V}^{(m)}_{\setminus\{i,j\}}$$

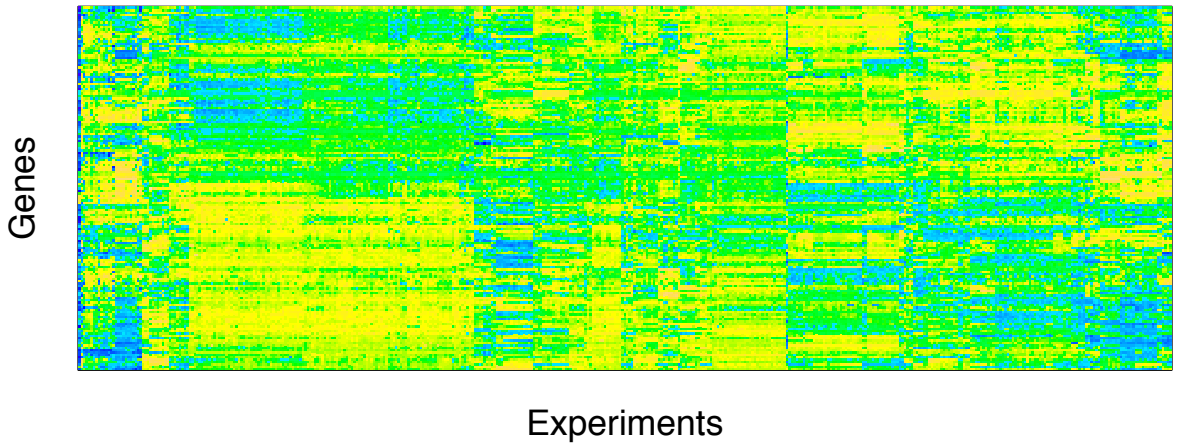
- Again, given measurements $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^{P}$ need to:
  - Select a test statistic
  - Construct an appropriate null distribution
  - Adjust for multiple testing

Case study: Inferring gene-regulatory interactions

- Genes are segments of DNA encoding information about cell functions
- Such information used in the expression of genes
  $\Rightarrow$ Creation of biochemical products, i.e., RNA or proteins
- Regulation of a gene refers to the control of its expression
  Ex: regulation exerted during transcription, copy of DNA to RNA
  $\Rightarrow$ Controlling genes are transcription factors (TFs)
  $\Rightarrow$ Controlled genes are termed targets
  $\Rightarrow$ Regulation type: activation or repression
- Regulatory interactions among genes basic to the workings of organisms
  $\Rightarrow$ Inference of interactions $\rightarrow$ Finding TF/target gene pairs
- Such relational information summarized in gene-regulatory networks

Regulatory interactions among E. coli genes

- Use microarray data and correlation methods to infer TF/target pairs



- Dataset: relative log expression RNA levels, for genes in E. coli
  - 4,345 genes measured under 445 different experimental conditions
- Ground truth: 153 TFs, and TF/target pairs from database RegulonDB

Methods to infer TF/target gene pairs

- Three correlation based methods to infer TF/target gene pairs

  $\Rightarrow$ Interactions declared if suitable p-values fall below a threshold

  Method 1: Pearson correlation between TF and potential target gene

  Method 2: Partial correlation, controlling for shared effects of one ($m = 1$) other TF, across all 152 other TFs

  Method 3: Full partial correlation, simultaneously controlling for shared effects of all ($m = 152$) other TFs

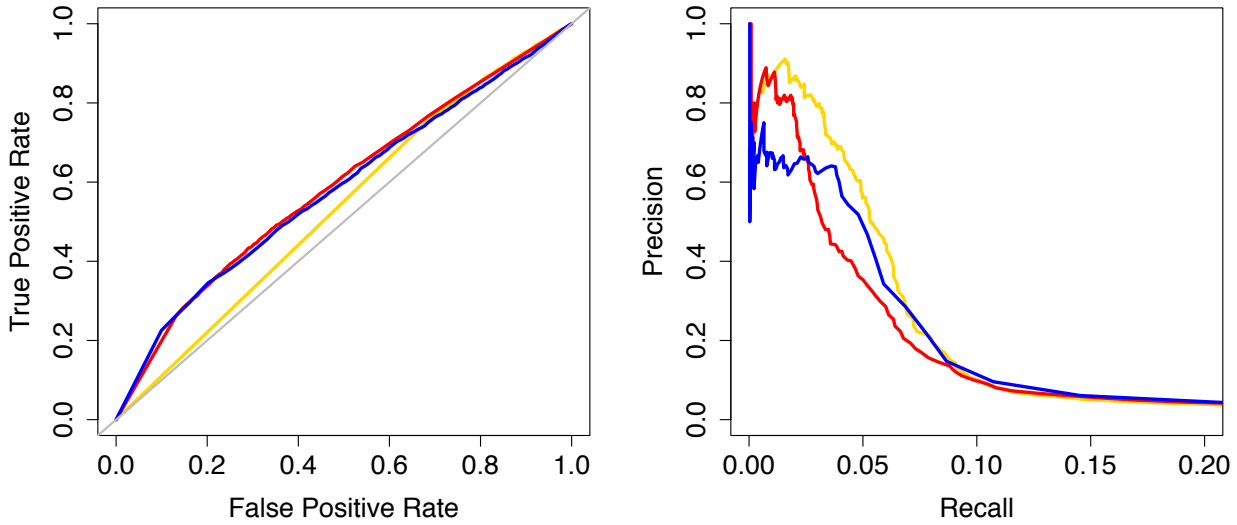
- In all cases applied Fisher transformation to obtain z-scores

  $\Rightarrow$ Asymptotic Gaussian distributions for p-values, with $P = 445$

- Compared inferred graphs to ground-truth network from RegulonDB

Performance comparisons

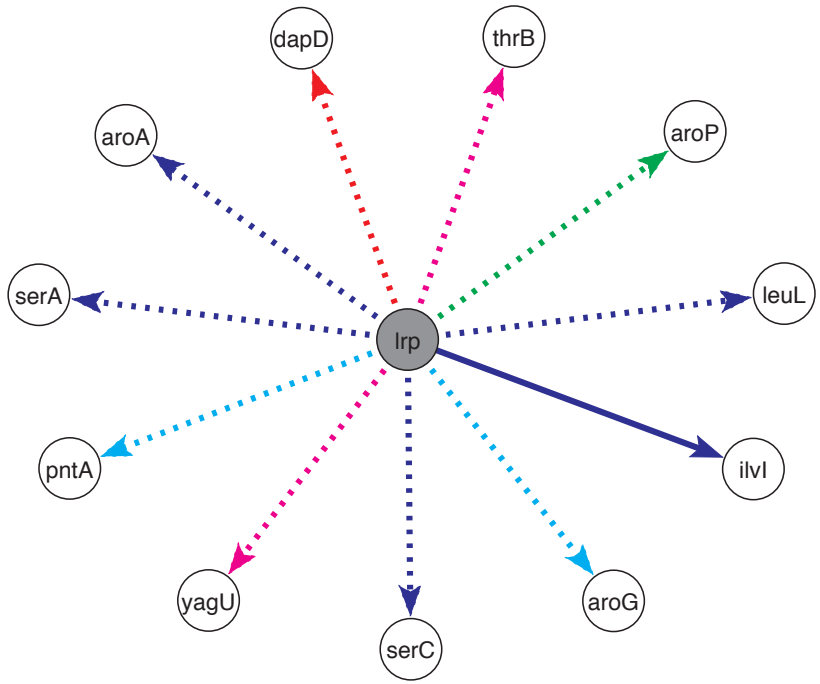
- ROC and Precision/Recall curves for Methods 1, 2, and 3
  - ⇒ Precision: fraction of predicted links that are true
  - ⇒ Recall: fraction of true links that are correctly predicted



- Method 1 performs worst, but none is stellar
  - ⇒ Correlation not strong indicator of regulation in this data
- All methods share a region of high precision, but a very small recall
  - ⇒ Limitations in number/diversity of profiles [Faith et al'07]

Predicting new TF/target gene pairs

- In biology, often interest is in predicting new interactions



- 11 interactions found for TF *lrp*, 10 experimentally confirmed (dotted)
  - $\Rightarrow$ 5 interacting target genes were new (magenta, red, cyan)
  - $\Rightarrow$ 4 present in RegulonDB (magenta, cyan), but not as *lrp* targets

Undirected Gaussian graphical models

- Suppose variables $\{x_i\}_{i\in\mathcal{V}}$ have multivariate Gaussian distribution

  $\Rightarrow$ Consider $\rho_{ij|\mathcal{V}\setminus\{i,j\}}$ conditioning on all other vertices ($m = N-2$)

Theorem

Under the Gaussian assumption, vertices $i, j \in \mathcal{V}$ have partial correlation

$$\rho_{ij|\mathcal{V}\setminus\{i,j\}} = 0$$

if and only if x_i and x_j are conditionally independent given $\{x_k\}_{k\in\mathcal{V}\setminus\{i,j\}}$

- **Def:** the conditional independence graph $G(\mathcal{V}, \mathcal{E})$ has edge set

$$\mathcal{E} = \left\{(i,j) \in \mathcal{V}^{(2)} : \rho_{ij|\mathcal{V}\setminus\{i,j\}} \neq 0\right\}$$

  $\Rightarrow$ A special and popular case of partial correlation networks

- Also known as Gaussian Markov random field (GMRF)

Covariance selection

- Let $\mathbf{\Sigma}$ be the covariance matrix of $\mathbf{x} = [x_1, \ldots, x_N]^\top$
  Def: the precision matrix is $\mathbf{\Theta} := \mathbf{\Sigma}^{-1}$ with entries θ_{ij}

- Key result: For GMRFs, the partial correlations can be expressed as

$$\rho_{ij|V\setminus\{i,j\}} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}}$$

  $\Rightarrow$ Non-zero entries in $\mathbf{\Theta} \Leftrightarrow$ Edges in the graph G

- Inferring G from $\mathcal{X}$ known as covariance selection [Dempster'74]
  $\Rightarrow$ Classical methods are 'network-agnostic,' and effectively test

$$H_0 : \rho_{ij|\mathcal{V}\setminus\{i,j\}} = 0 \quad \text{versus} \quad H_1 : \rho_{ij|\mathcal{V}\setminus\{i,j\}} \neq 0$$

- Often not scalable, and $P \ll N$ so estimation of $\hat{\mathbf{\Sigma}}$ challenging

Graphical Lasso

- Sparsity-regularized maximum-likelihood estimator of $\boldsymbol{\Theta}$ [Yuan-Lin'07]

$$\hat{\boldsymbol{\Theta}} \in \arg\max_{\boldsymbol{\Theta} \succeq \mathbf{0}} \left\{ \log\det \boldsymbol{\Theta} - \operatorname{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}) - \lambda \|\boldsymbol{\Theta}\|_1 \right\}$$

  $\Rightarrow$ Effective when $P \ll N$, encourages interpretable models

  $\Rightarrow$ Scalable solvers using coordinate-descent [Friedman et al'08]

- Performance guarantee: Graphical lasso with $\lambda = 2\sqrt{\frac{\log N}{P}}$ satisfies

$$\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_0\|_2 \leq \sqrt{\frac{d_{\max}^2 \log N}{P}} \quad \text{w.h.p.}$$

  $\Rightarrow$ Ground-truth $\boldsymbol{\Theta}_0$, maximum nodal degree $d_{\max}$

- Support consistency for $P = \Omega(d_{\max}^2 \log N)$ [Ravikumar et al'11]

GMRFs with Laplacian constraints

- Graphical model selection with Laplacian constraints $\mathbf{\Theta} = \mathbf{L}$
  - Off-diagonal entries $\theta_{ij} = L_{ij} = -A_{ij} \leq 0 \;\Rightarrow$ Attractive GMRF
  - Laplacian is singular ($\mathbf{L1} = \mathbf{0}$) $\Rightarrow$ Improper GMRF
- Estimate a proper GMRF via diagonal loading [Lake-Tenembaum'07]

$$\max_{\mathbf{\Theta} \succeq \mathbf{0}, \gamma \geq 0} \Big\{ \log\det \mathbf{\Theta} - \text{trace}(\hat{\mathbf{\Sigma}}\mathbf{\Theta}) - \lambda \|\mathbf{\Theta}\|_1 \Big\}$$
$$\text{s. to } \mathbf{\Theta} = \mathbf{L} + \gamma \mathbf{I}$$
$$\mathbf{L1} = \mathbf{0},\; L_{ij} \leq 0,\; i \neq j$$

$\Rightarrow$ Interpret γ^{-1} as variance of Gaussian isotropic fluctuations

- Favors graphs over which the signals are smooth (more later)

$$\text{trace}(\hat{\mathbf{\Sigma}}\mathbf{L}) \propto \sum_{p=1}^{P} \mathbf{x}_p^\top \mathbf{L} \mathbf{x}_p = \sum_{p=1}^{P} \text{TV}(\mathbf{x}_p)$$

Covariance selection meets linear regression

- **Idea:** separately estimate neighborhoods $\mathcal{N}_i := \{j : (i,j) \in \mathcal{E}\}$, $i \in \mathcal{V}$
- Conditional mean of x_i given $\mathbf{x}_{\setminus i} := [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N]^\top$ is

$$\mathbb{E}\left[x_i \,\middle|\, \mathbf{x}_{\setminus i}\right] = \mathbf{x}_{\setminus i}^\top \boldsymbol{\beta}^{(i)}$$

- Entries of $\boldsymbol{\beta}^{(i)}$ expressible in terms of those in $\hat{} = \boldsymbol{\Sigma}^{-1}$, namely

$$\beta_j^{(i)} = -\frac{\theta_{ij}}{\theta_{ii}}$$

  $\Rightarrow$ Non-zero $\beta_j^{(i)} \Leftrightarrow$ Non-zero θ_{ij} in $\hat{} \Leftrightarrow$ Edge (i,j) in G

  $\Rightarrow$ In other words, $\text{supp}(\boldsymbol{\beta}^{(i)}) := \{j : \beta_j^{(i)} \neq 0\} \equiv \mathcal{N}_i$

- Suggests inference of G via least-squares (LS) regression, since

$$\boldsymbol{\beta}^{(i)} = \arg\min_{\boldsymbol{\beta}} \mathbb{E}\left[(x_i - \mathbf{x}_{\setminus i}^\top \boldsymbol{\beta})^2\right], \quad i \in \mathcal{V}$$

Neighborhood-based sparse regression

- Cycle over vertices $i \in \mathcal{V}$ and estimate $\hat{\mathcal{N}}_i = \text{supp}(\hat{\boldsymbol{\beta}}^{(i)})$, where

$$\hat{\boldsymbol{\beta}}^{(i)} \in \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{N-1}} \left\{ \sum_{p=1}^{P} (x_{pi} - \mathbf{x}_{p,\setminus i}^{\top} \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}$$

  $\Rightarrow$ Separable lasso problems per vertex

- No guarantee that $\hat{\beta}_j^{(i)} \neq 0$ implies $\hat{\beta}_i^{(j)} \neq 0$ and vice versa

  $\Rightarrow$ Combine information in $\hat{\mathcal{N}}_i$ and $\hat{\mathcal{N}}_j$ to enforce symmetry

  $\Rightarrow$ OR rule: $(i,j) \in \mathcal{E}$ if $\beta_j^{(i)} \neq 0$ or $\beta_i^{(j)} \neq 0$. Likewise, AND rule

- Support consistency for either rule [Meinshausen-Bühlmann'06]
  - Suitable choice of λ, sparsity of $\boldsymbol{\Theta}_0$, and sample complexity $P \ll N$

Conceptual roadmap for GMRF model selection

Testing partial correlations

For each $(i,j) \in \mathcal{V} \times \mathcal{V}$, test the hypothesis

$$H_0 : \rho_{ij|\mathcal{V}\setminus ij} = 0 \quad \text{versus} \quad H_1 : \rho_{ij|\mathcal{V}\setminus ij} \neq 0$$

Covariance selection

$$\rho_{ij|\mathcal{V}\setminus ij} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}} \rightarrow \rho_{ij|\mathcal{V}\setminus ij} \neq 0 \Leftrightarrow \theta_{ij} \neq 0$$

Infer non-zero entries $\theta_{ij} \neq 0$ of the precision matrix

$$\boldsymbol{\Theta} := \boldsymbol{\Sigma}^{-1}$$

Neighborhood-based regression

$$\beta_j^{(i)} = -\frac{\theta_{ij}}{\theta_{ii}} \rightarrow \beta_j^{(i)} \neq 0 \Leftrightarrow \theta_{ij} \neq 0$$

For each $i \in \mathcal{V}$, infer non-zero regression coefficients $\beta_j^{(i)} \neq 0$ in

$$\boldsymbol{\beta}^{(i)} = \arg\min_{\boldsymbol{\beta}} \mathbb{E}\left[(x_i - \mathbf{x}_{\setminus i}^T \boldsymbol{\beta})^2\right]$$

Comparative summary

- Parallelizable neighborhood-based regression (NBR)
  - $\Rightarrow$ Conditional likelihood per vertex $i \in \mathcal{V}$, disregards $\boldsymbol{\Theta} \succeq \mathbf{0}$
  - $\Rightarrow$ Tends to be computationally faster
- Graphical Lasso minimizes a (regularized) global likelihood

$$\mathcal{L}(\boldsymbol{\Theta}; \mathcal{X}) = \log\det\boldsymbol{\Theta} - \text{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})$$

  - $\Rightarrow$ Tends to be (statistically) more efficient
- NBR method tractable even for discrete or mixed graphical models
  - $\Rightarrow$ Ising-model selection for $\mathbf{x} \in \{-1, +1\}^N$ [Ravikumar'10]

Learning graphs from smooth signals

Statistical methods for network topology inference

Learning graphs from observations of smooth signals

Identifying the structure of network diffusion processes

Problem formulation

Rationale

- Seek graphs on which data admit certain regularities
  - Nearest-neighbor prediction (a.k.a. graph smoothing)
  - Semi-supervised learning
  - Efficient information-processing transforms
- Many real-world graph signals are smooth
  - Graphs based on similarities among vertex attributes
  - Network formation driven by homophily, proximity in latent space

Problem statement

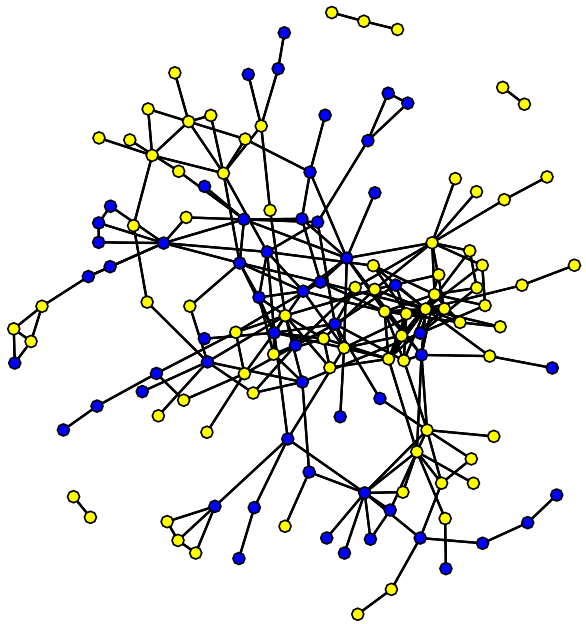
Given observations $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^{P}$, identify a graph G such that signals in $\mathcal{X}$ are smooth on G.

- Criterion: Dirichlet energy on the graph G with Laplacian $\mathbf{L}$

$$\text{TV}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L} \mathbf{x}$$

Example: Predicting protein function

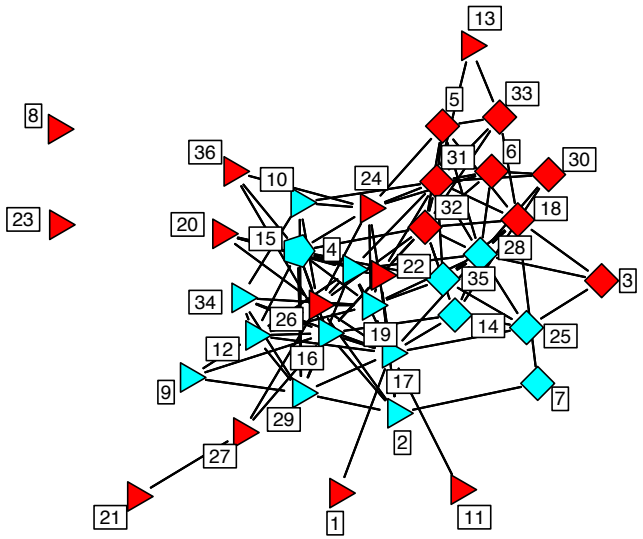
- Baker's yeast data, formally known as *Saccharomyces cerevisiae*
  - Graph: 134 vertices (proteins) and 241 edges (protein interactions)



- Signal: functional annotation intracellular signaling cascade (ICSC)
  - Signal transduction, how cells react to the environment
  - $x_i = 1$ if protein i annotated ICSC (yellow), $x_i = 0$ otherwise (blue)

Example: Predicting law practice

- Working relationships among lawyers [Lazega'01]
  - Graph: 36 partners, edges indicate partners worked together



- Signal: various node-level attributes $\mathbf{x} = \{x_i\}_{i \in \mathcal{V}}$ including

  $\Rightarrow$ Type of practice, i.e., litigation (red) and corporate (cyan)

- Suspect lawyers collaborate more with peers in same legal practice

  $\Rightarrow$ Knowledge of collaboration useful in predicting type of practice

Laplacian-based factor analysis model

- Consider an unknown graph G with Laplacian $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$
  $\Rightarrow$ Adopt GFT basis $\mathbf{V}$ as signal representation matrix
- Factor-analysis model for the observed graph signal [Dong et al'16]

$$\mathbf{x} = \mathbf{V}\boldsymbol{\chi} + \boldsymbol{\epsilon}$$

  $\Rightarrow$ Latent variables $\boldsymbol{\chi} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^\dagger)$ ($\approx$ GFT coefficients)
  $\Rightarrow$ Isotropic error term $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$
- Smoothness: prior encourages low-pass bandlimited $\mathbf{x}$
  $\Rightarrow$ Small eigenvalues of $\mathbf{L}$ (low freq.) $\rightarrow$ High-power factor loadings

Inference as denoising via graph kernel regression

- Maximum a posteriori (MAP) estimator of the latent variables $\boldsymbol{\chi}$

$$\hat{\boldsymbol{\chi}}_{\text{MAP}} = \arg\min_{\boldsymbol{\chi}} \left\{ \|\mathbf{x} - \mathbf{V}\boldsymbol{\chi}\|^2 + \alpha \boldsymbol{\chi}^\top \boldsymbol{\Lambda} \boldsymbol{\chi} \right\}$$

  $\Rightarrow$ Parameterized by the unknown $\mathbf{V}$ and $\boldsymbol{\Lambda}$

- Define predictor $\mathbf{y} := \mathbf{V}\boldsymbol{\chi}$, regularizer expressible as

$$\boldsymbol{\chi}^\top \boldsymbol{\Lambda} \boldsymbol{\chi} = \mathbf{y}^\top \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^\top \mathbf{y} = \mathbf{y}^\top \mathbf{L} \mathbf{y} = \text{TV}(\mathbf{y})$$

  $\Rightarrow$ Laplacian-based TV denoiser of $\mathbf{x}$, smoothness prior on $\mathbf{y}$

  $\Rightarrow$ Kernel-ridge regression with unknown $\mathbf{K} := \mathbf{L}^\dagger$ (graph filter)

- **Idea:** jointly search for $\mathbf{L}$ and denoised representation $\mathbf{y} = \mathbf{V}\boldsymbol{\chi}$

$$\min_{\mathbf{L}, \mathbf{y}} \left\{ \|\mathbf{x} - \mathbf{y}\|^2 + \alpha \mathbf{y}^\top \mathbf{L} \mathbf{y} \right\}$$

Formulation and algorithm

- Given signals $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$ in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, solve

$$\min_{\mathbf{L},\mathbf{Y}} \left\{ \|\mathbf{X} - \mathbf{Y}\|_F^2 + \alpha \text{trace}\left(\mathbf{Y}^\top \mathbf{L}\mathbf{Y}\right) + \frac{\beta}{2}\|\mathbf{L}\|_F^2 \right\}$$

$$\text{s. to} \quad \text{trace}(\mathbf{L}) = N,\ \mathbf{L}\mathbf{1} = \mathbf{0},\ L_{ij} = L_{ji} \leq 0,\ i \neq j$$

  $\Rightarrow$ Objective function: Fidelity + smoothness + edge sparsity

  $\Rightarrow$ Not jointly convex in $\mathbf{L}$ and $\mathbf{Y}$, but bi-convex

- Algorithmic approach: alternating minimization (AM), $O(N^3)$ cost

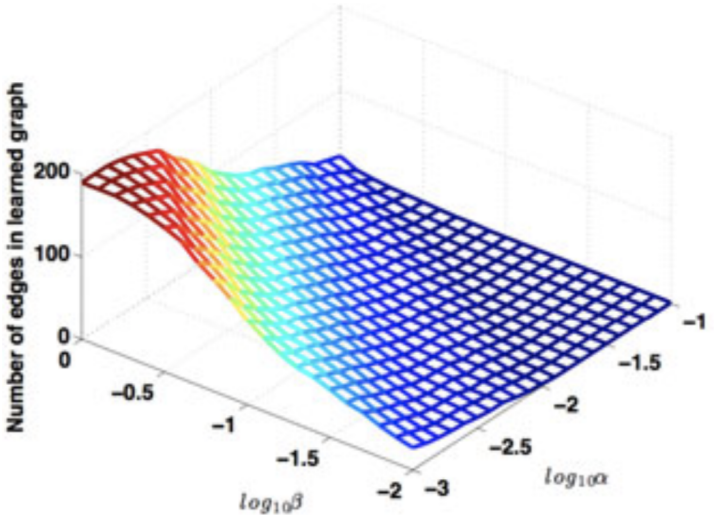
  (S1) Fixed $\mathbf{Y}$: solve for $\mathbf{L}$ via interior-point method, ADMM (more soon)

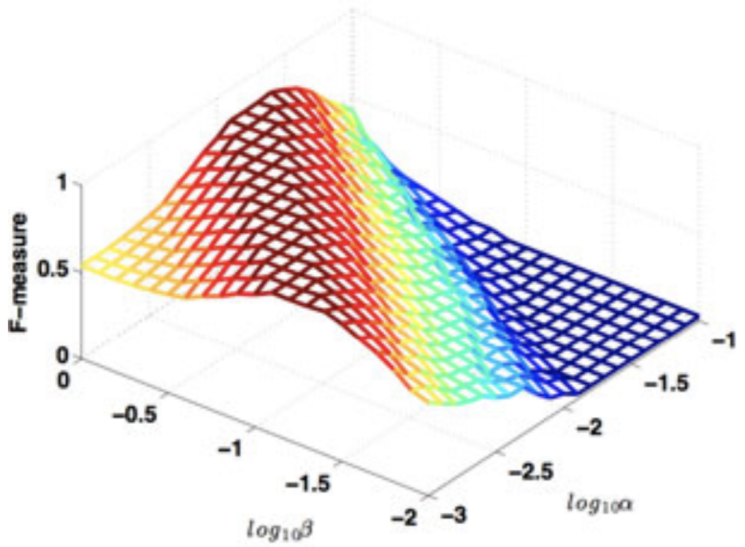
  (S2) Fixed $\mathbf{L}$: low-pass, graph filter-based smoother of the signals in $\mathbf{X}$

$$\mathbf{Y} = (\mathbf{I} + \alpha\mathbf{L})^{-1}\mathbf{X}$$

Impact of regularizers on sparsity and accuracy

- Generate multiple signals on a synthetic Erdős-Rényi graph
  $\Rightarrow$ Recover the graph for different values of α and β

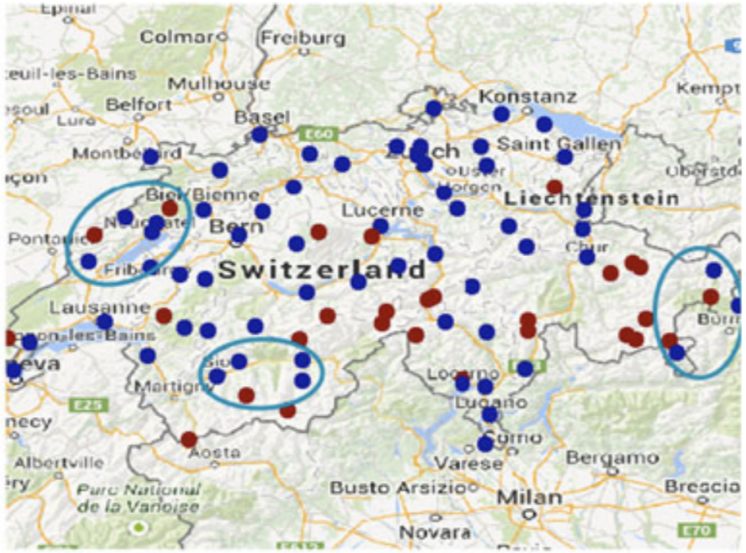




- More edges promoted by increasing β and decreasing α
- In the low noise regime, the ratio β/α determines behavior

Example: Temperature graph in Switzerland

- $N = 89$ stations measuring monthly temperature averages (1981-2010)
  $\Rightarrow$ Learn a graph G on which the temperatures vary smoothly
- Geographical distance not a good idea $\Rightarrow$ different altitudes



- Recover altitude partition from spectral clustering on G
  $\Rightarrow$ Red (high stations) and blue (low stations) clusters
- K-means applied directly to the temperatures (right) fails

Signal smoothness meets edge sparsity

- Recall $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, let $\bar{\mathbf{x}}_i^\top \in \mathbb{R}^{1 \times P}$ denote its i-th row
  $\Rightarrow$ Euclidean distance matrix $\mathbf{Z} \in \mathbb{R}_+^{N \times N}$, where $Z_{ij} := \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|^2$
- Neat trick: link between smoothness and sparsity [Kalofolias'16]

$$\sum_{p=1}^{P} \text{TV}(\mathbf{x}_p) = \text{trace}(\mathbf{X}^\top \mathbf{L} \mathbf{X}) = \frac{1}{2} \|\mathbf{A} \circ \mathbf{Z}\|_1$$

  $\Rightarrow$ Sparse $\mathcal{E}$ when data come from a smooth manifold
  $\Rightarrow$ Favor candidate edges (i, j) associated with small Z_{ij}
- Shows that edge sparsity on top of smoothness is redundant
- Parameterize graph learning problems in terms of $\mathbf{A}$ (instead of $\mathbf{L}$)
  $\Rightarrow$ Advantageous since constraints on $\mathbf{A}$ are decoupled

Scalable topology identification framework

- General purpose model for learning graphs [Kalofolias'16]

$$\min_{\mathbf{A}} \left\{ \|\mathbf{A} \circ \mathbf{Z}\|_1 - \alpha \mathbf{1}^\top \log(\mathbf{A}\mathbf{1}) + \frac{\beta}{2}\|\mathbf{A}\|_F^2 \right\}$$
$$\text{s. to} \quad \operatorname{diag}(\mathbf{A}) = \mathbf{0},\ A_{ij} = A_{ji} \geq 0,\ i \neq j$$

  $\Rightarrow$ Logarithmic barrier forces positive degrees

  $\Rightarrow$ Penalize large edge-weights to control sparsity

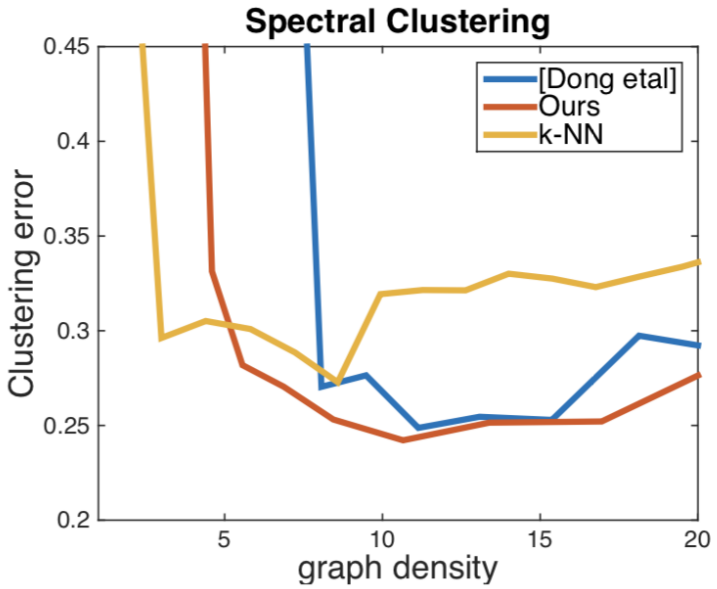
- Primal-dual solver amenable to parallelization, $O(N^2)$ cost

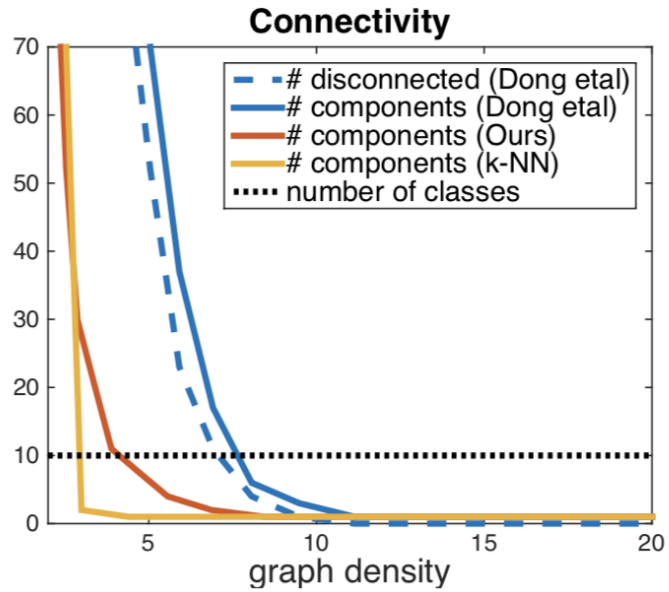
- Laplacian-based factor analysis encore. Tackle **(S1)** as

$$\min_{\mathbf{A}} \left\{ \|\mathbf{A} \circ \mathbf{Z}\|_1 - \log(\mathbb{I}\{\|\mathbf{A}\|_1 = N\}) + \frac{\beta}{2}\left(\|\mathbf{A}\mathbf{1}\|^2 + \|\mathbf{A}\|_F^2\right) \right\}$$
$$\text{s. to} \quad \operatorname{diag}(\mathbf{A}) = \mathbf{0},\ A_{ij} = A_{ji} \geq 0,\ i \neq j$$

Example: Learning the graph of USPS digits

- 1001 images of the 10 digits, but highly imbalanced ($2.6i^2$)
  - $\Rightarrow$ 10 classes via graph recovery plus spectral clustering
- Compare two methods based on smoothness and k-NN graph





- Performance more robust to graph density
  - $\Rightarrow$ Likely attributable to non-singleton nodes

Graph learning via edge subset selection

- **Idea:** parameterize the unknown topology via an edge indicator vector
- Complete graph on N nodes, having $M := \binom{N}{2}$ edges
  $\Rightarrow$ Incidence matrix $\mathbf{B} := [\mathbf{b}_1, \ldots, \mathbf{b}_M] \in \mathbb{R}^{N \times M}$
- Laplacian of a candidate graph $G(\mathcal{V}, \mathcal{E})$ [Chepuri et al'17]

$$\mathbf{L}(\boldsymbol{\omega}) = \sum_{m=1}^{M} \omega_m \mathbf{b}_m \mathbf{b}_m^\top$$

  $\Rightarrow$ Binary edge indicator vector $\boldsymbol{\omega} := [\omega_1, \ldots, \omega_M]^\top \in \{0, 1\}^M$
  $\Rightarrow$ Offers an explicit handle on the number of edges $\|\boldsymbol{\omega}\|_0 = |\mathcal{E}|$

Problem: Given observations $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$, learn an unweighted graph $G(\mathcal{V}, \mathcal{E})$ such that signals in $\mathcal{X}$ are smooth on G and $|\mathcal{E}| = K$.

Cardinality-constrained Boolean optimization

- Natural formulation is to solve the non-convex problem

$$\min_{\boldsymbol{\omega}\in\{0,1\}^M} \text{trace}(\mathbf{X}^\top \mathbf{L}(\boldsymbol{\omega})\mathbf{X}), \quad \text{s. to } \|\boldsymbol{\omega}\|_0 = K$$

- Solution obtained through a simple rank-ordering procedure
  - Compute edge scores $c_m := \text{trace}(\mathbf{X}^\top(\mathbf{b}_m\mathbf{b}_m^\top)\mathbf{X})$
  - Set $\omega_m = 1$ for those K edges having the smallest scores

- More pragmatic AWGN setting where $\mathbf{x}_p = \mathbf{y}_p + \boldsymbol{\epsilon}_p$, $p = 1, \ldots, P$

$$\min_{\mathbf{Y},\boldsymbol{\omega}\in\{0,1\}^M} \left\{\|\mathbf{X}-\mathbf{Y}\|_F^2 + \alpha\text{trace}(\mathbf{Y}^\top\mathbf{L}(\boldsymbol{\omega})\mathbf{Y})\right\}, \quad \text{s. to } \|\boldsymbol{\omega}\|_0 = K$$

$\Rightarrow$ Tackle via AM or semidefinite relaxation (SDR)

Comparative summary

- Noteworthy features of the edge subset selection approach
  - ✓ Direct control on edge sparsity
  - ✓ Simple algorithm in the noise-free case
  - ✓ Devoid of Laplacian feasibility constraints
  - ✗ Does not guarantee connectivity of G
  - ✗ No room for optimizing edge weights

- Scalable framework in [Kalofolias'16] also quite flexible

$$\min_{\mathbf{A}} \{\|\mathbf{A} \circ \mathbf{Z}\|_1 + g(\mathbf{A})\}$$
$$\text{s. to} \quad \text{diag}(\mathbf{A}) = \mathbf{0},\ A_{ij} = A_{ji} \geq 0,\ i \neq j$$

  $\Rightarrow$ Subsumes the factor-analysis model [Dong et al'16]

  $\Rightarrow$ Recovers Gaussian kernel weights $A_{ij} := \exp\left(-\frac{\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|^2}{\sigma^2}\right)$ for

$$g(\mathbf{A}) = \sigma^2 \sum_{i,j} A_{ij}(\log(A_{ij}) - 1)$$

Case study: Classification of network data

- Labeled graph signals $\mathcal{X}_c := \{\mathbf{x}_p^{(c)}\}_{p=1}^{P_c}$ from C different classes
  - $\Rightarrow$ Signals in each class possess a very distincitve structure
- **As.:** Class c signals are smooth w.r.t. unknown $G_c(\mathcal{V}, \mathcal{E}_c)$
- Multiple linear subspace model
  - $\Rightarrow$ Signals spanned by few Laplacian modes (GFT components)
  - $\Rightarrow$ Like susbpace clustering [Vidal'11], but with supervision

Problem statement

Given training signals $\mathcal{X} = \bigcup_{c=1}^{C} \mathcal{X}_c$, learn discriminative graphs $\mathbf{A}_c$ under smoothness priors to classify test signals via GFT projections.

Discriminative graph learning

- Discriminative graph learning per class c [Saboksayr et al'21]

$$\min_{\mathbf{A}_c} \left\{ \|\mathbf{A}_c \circ \mathbf{Z}_c\|_1 - \alpha \mathbf{1}^\top \log(\mathbf{A}_c \mathbf{1}) + \frac{\beta}{2}\|\mathbf{A}_c\|_F^2 - \gamma \sum_{k \neq c}^{C} \|\mathbf{A}_c \circ \mathbf{Z}_k\|_1 \right\}$$

s. to $\quad \text{diag}(\mathbf{A}_c) = \mathbf{0},\ [\mathbf{A}_c]_{ij} = [\mathbf{A}_c]_{ji} \geq 0,\ i \neq j$

$\Rightarrow$ Capture the underlying graph topology (class c structure)

$\Rightarrow$ Discriminability to boost classification performance

- **Q:** Given graphs $\{\hat{\mathbf{A}}_c\}_{c=1}^{C}$, how do we classify a test signal $\mathbf{x}$?

- Pass $\mathbf{x}$ through a filter-bank with C low-pass filters (LPFs)

$$\tilde{\mathbf{x}}_{F,c} = \text{diag}(\tilde{\mathbf{h}})\hat{\mathbf{V}}_c^\top \mathbf{x} \quad \Rightarrow \quad \hat{c} = \underset{c}{\text{argmax}} \left\{ \|\tilde{\mathbf{x}}_{F,c}\|^2 \right\}$$

$\Rightarrow$ LPF frequency response $\tilde{\mathbf{h}}$, learned class-c GFT basis $\hat{\mathbf{V}}_c$

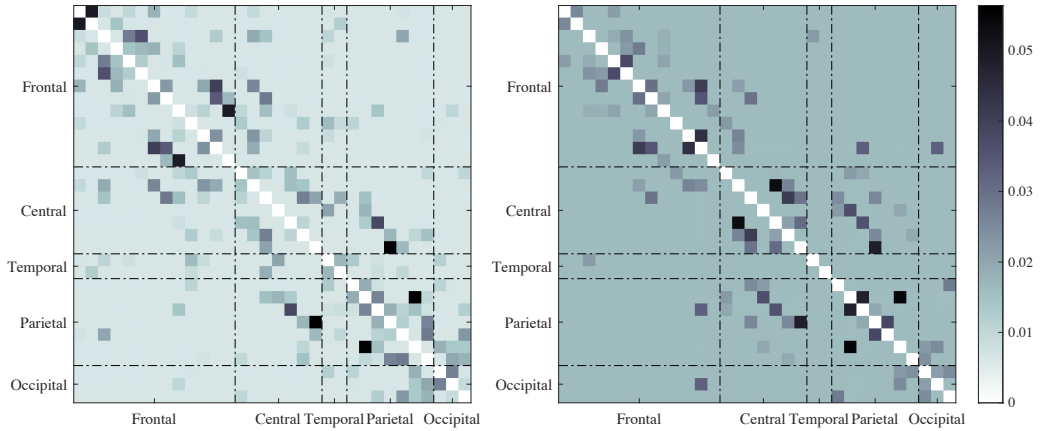
Emotion recognition from EEG records

- Discriminative graph learning for emotion recognition from EEG signals
- DEAP dataset $\Rightarrow$ 32 subjects watch music videos (40 trials each)
  - Asked to rate videos: valence, arousal, like/dislike, dominance
  - Focus on valence labels: low (1-5 rating) and high (6-10 rating)
  - Signals acquired from $N = 32$ EEG channels
- We perform a subject-specific valence classification task
  - $\Rightarrow$ Learn $C = 2$ graphs and project onto the 8 smoothest modes
  - $\Rightarrow$ Report leave-one (trial)-out classification accuracy
- Mean classification accuracy over subjects is 92.73%

S. S. Saboksayr et al, "Online discriminative graph learning from multi-class smooth signals," *arXiv:2101.00184 [eess.SP]*, 2021

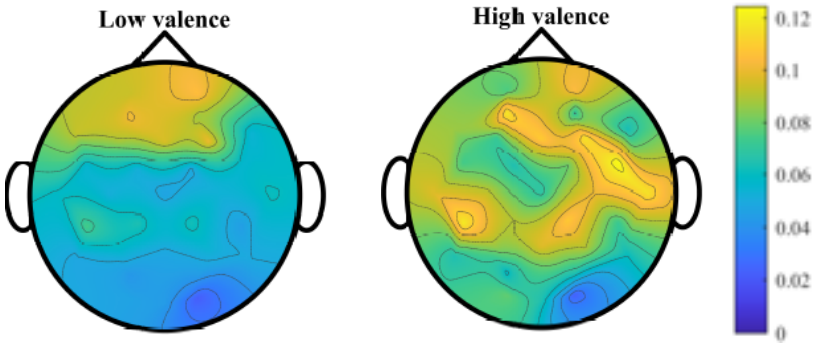
Valence classification

- **Q:** What information do we glean from the class-conditional graphs?



Low valence | High valence | Different connections (p=0.002)

- Connectivity increases with emotion intensity (frontal lobe links)



- Asymmetric frontal activity apparent from the 8 smoothest modes

Learning graphs from diffused signals

Statistical methods for network topology inference

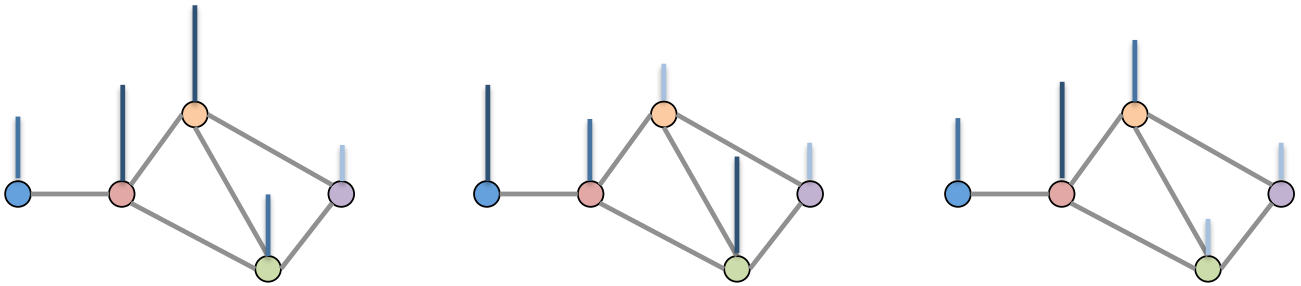
Learning graphs from observations of smooth signals

Identifying the structure of network diffusion processes

Problem formulation

Setup

- Undirected network G with unknown graph shift $\mathbf{S}$
- Observe signals $\{\mathbf{y}_i\}_{i=1}^P$ defined on the unknown graph



$\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{y}_3$

Problem statement

Given observations $\{\mathbf{y}_i\}_{i=1}^P$, determine the network $\mathbf{S}$ knowing that $\{\mathbf{y}_i\}_{i=1}^P$ are outputs of a diffusion process on $\mathbf{S}$.

Generating structure of a diffusion process

- Signal $\mathbf{y}_i$ is the response of a linear diffusion process to input $\mathbf{x}_i$

$$\mathbf{y}_i \;=\; \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x}_i \;=\; \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}_i, \quad i = 1, \ldots, P$$

  $\Rightarrow$ Common generative model, e.g., heat diffusion, consensus

- Cayley-Hamilton asserts we can write diffusion as ($L \leq N$)

$$\mathbf{y}_i = \left(\sum_{l=0}^{L-1} h_l \mathbf{S}^l \right) \mathbf{x}_i := \mathbf{H}\mathbf{x}_i, \quad i = 1, \ldots, P$$

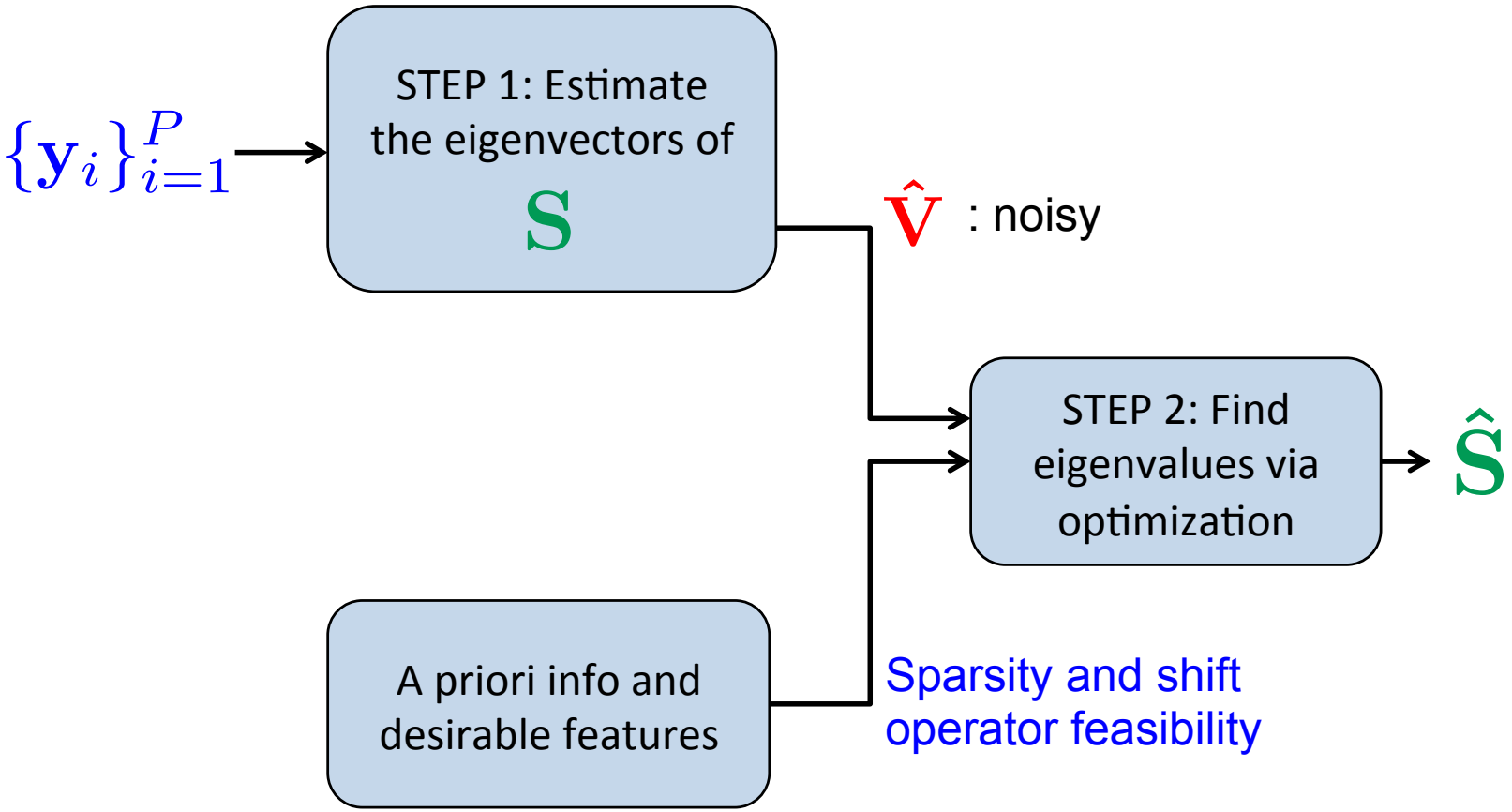
  $\Rightarrow$ Graph filter $\mathbf{H}$ is shift invariant [Sandryhaila-Moura'13]

  $\Rightarrow$ $\mathbf{H}$ diagonalized by the eigenvectors $\mathbf{V}$ of the shift operator

- **Goal**: estimate undirected network $\mathbf{S}$ from signal realizations $\{\mathbf{y}_i\}_{i=1}^P$

  $\Rightarrow$ Unknowns: filter order L, coefficients $\{h_l\}_{l=1}^{L-1}$, inputs $\{\mathbf{x}_i\}_{i=1}^P$

Blueprint of our solution



Step 1: Obtaining the eigenvectors of $\mathbf{S}$

- $\mathbf{y}$ is the output of a local diffusion of a white input

$$\mathbf{y} = \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x} = \left(\sum_{l=0}^{N-1} h_l \mathbf{S}^l \right) \mathbf{x} := \mathbf{H}\mathbf{x}$$

- The covariance $\mathbf{C}_y$ of $\mathbf{y}$ shares $\mathbf{V}$ with $\mathbf{S}$

$$\mathbf{C}_y = \mathbf{H}^2 = h_0^2 \mathbf{I} + 2h_0 h_1 \mathbf{S} + h_1^2 \mathbf{S}^2 + \ldots$$

- Mapping $\mathbf{S} \to \mathbf{C}_y$ is polynomial
  - $\Rightarrow$ Correlation methods $\Rightarrow \mathbf{C}_y = \mathbf{S}$
  - $\Rightarrow$ Precision methods (graphical Lasso) $\to \mathbf{C}_y = \mathbf{S}^{-1}$
  - $\Rightarrow$ Structural EM methods $\Rightarrow \mathbf{C}_y = (\mathbf{I} - \mathbf{S})^{-2}$

Correlated input signals

- **Q:** What if the signal $\mathbf{x}$ is colored?

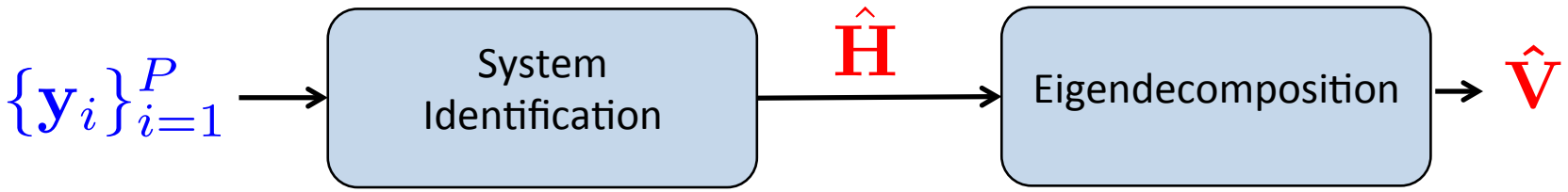
  $\Rightarrow$ Matrices $\mathbf{S}$ and $\mathbf{C}_y$ no longer simultaneously diagonalizable since

$$\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$$

- Key: still $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$ diagonalized by the eigenvectors $\mathbf{V}$ of $\mathbf{S}$

  $\Rightarrow$ Infer $\mathbf{V}$ by estimating the unknown diffusion (graph) filter $\mathbf{H}$

  $\Rightarrow$ Step 1 boils down to system identification + eigendecomposition



- Henceforth assume $\mathbf{C}_x$ is non-singluar and known

System ID as matrix quadratic equation

- **Q:** What are the solutions of the quadratic equation $\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$?

Proposition: Define $\mathbf{C}_{xyx} := \mathbf{C}_x^{1/2}\mathbf{C}_y\mathbf{C}_x^{1/2}$, with eigenvectors $\mathbf{V}_{xyx}$. Then all admissible symmetric graph filters $\mathbf{H}$ are of the form

$$\mathbf{H} = \mathbf{C}_x^{-1/2}\mathbf{C}_{xyx}^{1/2}\mathbf{V}_{xyx}\text{diag}(\mathbf{b})\mathbf{V}_{xyx}^\top\mathbf{C}_x^{-1/2},$$

where $\mathbf{b} \in \{-1, 1\}^N$ is a binary (signed) vector.

- Even if we know $\mathbf{C}_y$ perfectly, $\mathbf{H}$ is not identifiable
  - $\Rightarrow$ Not surprising since we only have second-moment information
  - $\Rightarrow$ Unique solution $\mathbf{H} = \mathbf{C}_x^{-1/2}\mathbf{C}_{xyx}^{1/2}\mathbf{C}_x^{-1/2}$ for positive semidefinite $\mathbf{H}$
- Consider having access to multiple input distributions $\{\mathbf{C}_{x,m}\}_{m=1}^M$

Boolean quadratic program

- Define $\mathbf{A}_m := (\mathbf{C}_{x,m}^{-1/2}\mathbf{V}_{xyx,m}) \odot (\mathbf{C}_{x,m}^{-1/2}\mathbf{C}_{xyx,m}^{1/2}\mathbf{V}_{xyx,m})$ and form

$$\boldsymbol{\Psi} := \begin{bmatrix} \mathbf{A}_1 & -\mathbf{A}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & -\mathbf{A}_3 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{M-1} & -\mathbf{A}_M \end{bmatrix}$$

- With $\mathbf{b}_m \in \{-1,1\}^N$ and $\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T, \ldots, \mathbf{b}_M^T]^\top$, then $\boldsymbol{\Psi}\mathbf{b}^* = \mathbf{0}$
- In practice only $\{\hat{\mathbf{C}}_{y,m}\}_{m=1}^M$ are available $\Rightarrow$ Estimate $\mathbf{b}^*$ as

$$\hat{\mathbf{b}}^* = \underset{\mathbf{b}\in\{-1,1\}^{NM}}{\operatorname{argmin}} \; \mathbf{b}^\top \hat{\boldsymbol{\Psi}}^\top \hat{\boldsymbol{\Psi}} \mathbf{b}$$

- Solution $\hat{\mathbf{b}}^*$ of binary quadratic program (BQP) $\Rightarrow$ Filter estimate

$$\hat{\mathbf{H}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{C}_{x,m}^{-1/2} \hat{\mathbf{C}}_{xyx,m}^{1/2} \hat{\mathbf{V}}_{xyx,m} \operatorname{diag}(\hat{\mathbf{b}}_m^*) \hat{\mathbf{V}}_{xyx,m}^\top \mathbf{C}_{x,m}^{-1/2}$$

Semidefinite relaxation

- System identification reduces to solving the NP-hard BQP

$$\hat{\mathbf{b}}^* = \underset{\mathbf{b}\in\{-1,1\}^{NM}}{\operatorname{argmin}} \; \mathbf{b}^\top \hat{\boldsymbol{\Psi}}^\top \hat{\boldsymbol{\Psi}} \mathbf{b}$$

- Define $\hat{\mathbf{W}} = \hat{\boldsymbol{\Psi}}^\top \hat{\boldsymbol{\Psi}}$ and $\mathbf{B} = \mathbf{b}\mathbf{b}^\top$, BQP equivalent to

$$\min_{\mathbf{B}\succeq\mathbf{0}} \operatorname{tr}(\hat{\mathbf{W}}\mathbf{B}) \quad \text{s. to} \quad \operatorname{rank}(\mathbf{B}) = 1, \; B_{ii} = 1, \; i = 1, \ldots, NM$$

- Drop source of non-convexity $\Rightarrow$ Semidefinite relaxation (SDR)

$$\mathbf{B}^* = \underset{\mathbf{B}\succeq\mathbf{0}}{\operatorname{argmin}} \operatorname{tr}(\hat{\mathbf{W}}\mathbf{B}) \quad \text{s. to} \quad B_{ii} = 1, \; i = 1, \ldots, NM$$

Performance guarantee

- For $l = 1, \ldots, L$, draw $\mathbf{z}_l \sim \mathcal{N}(\mathbf{0}, \mathbf{B}^*)$, round $\tilde{\mathbf{b}}_l = \text{sign}(\mathbf{z}_l)$, to obtain

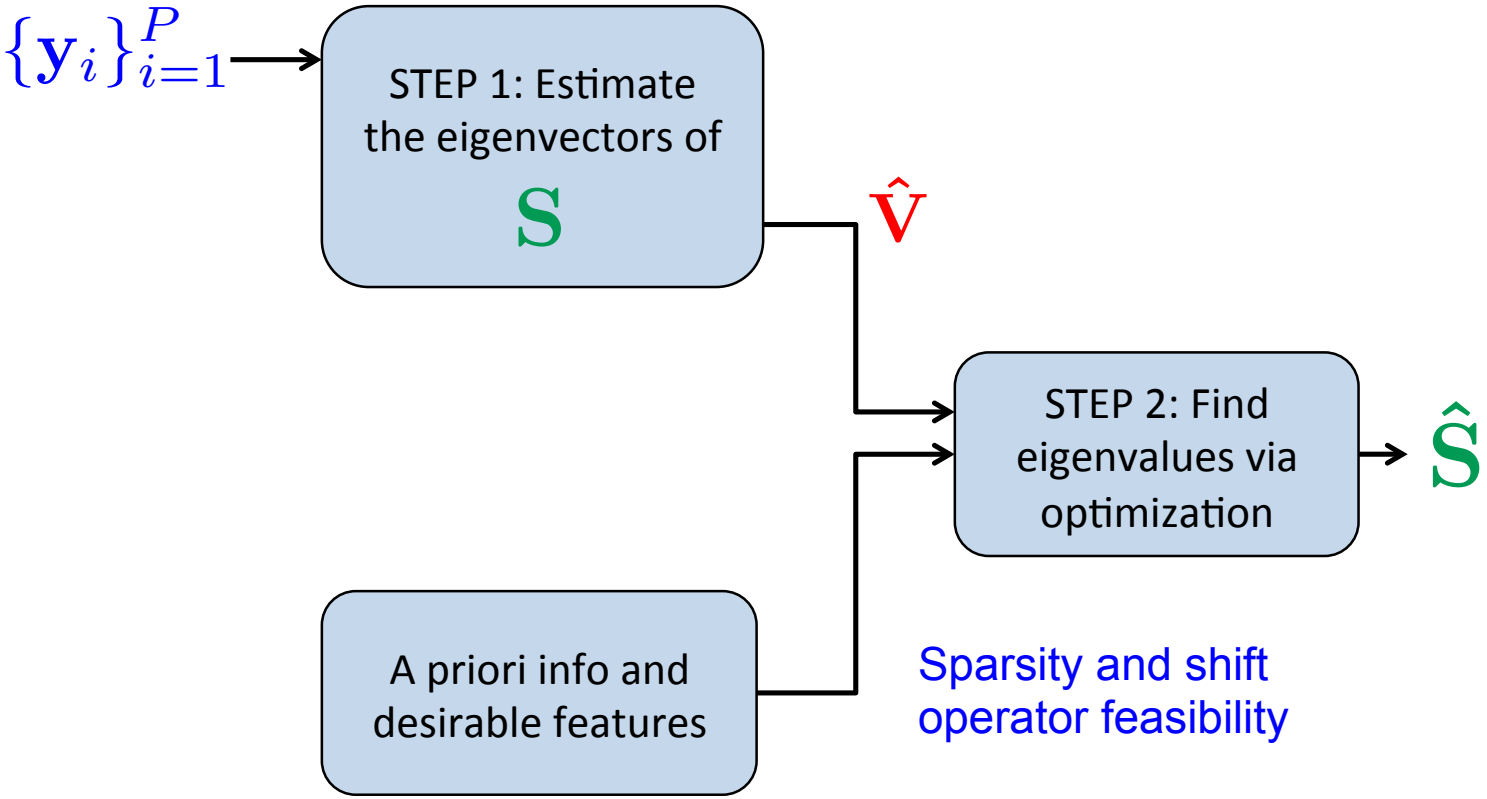
$$l^* = \underset{l=1,\ldots,L}{\text{argmin}}\, \tilde{\mathbf{b}}_l^T \hat{\mathbf{W}} \tilde{\mathbf{b}}_l$$

Theorem: Let $\hat{\mathbf{b}}^*$ be the BQP solution and $\tilde{\mathbf{b}}_{l^*}$ the SDR output. Then,

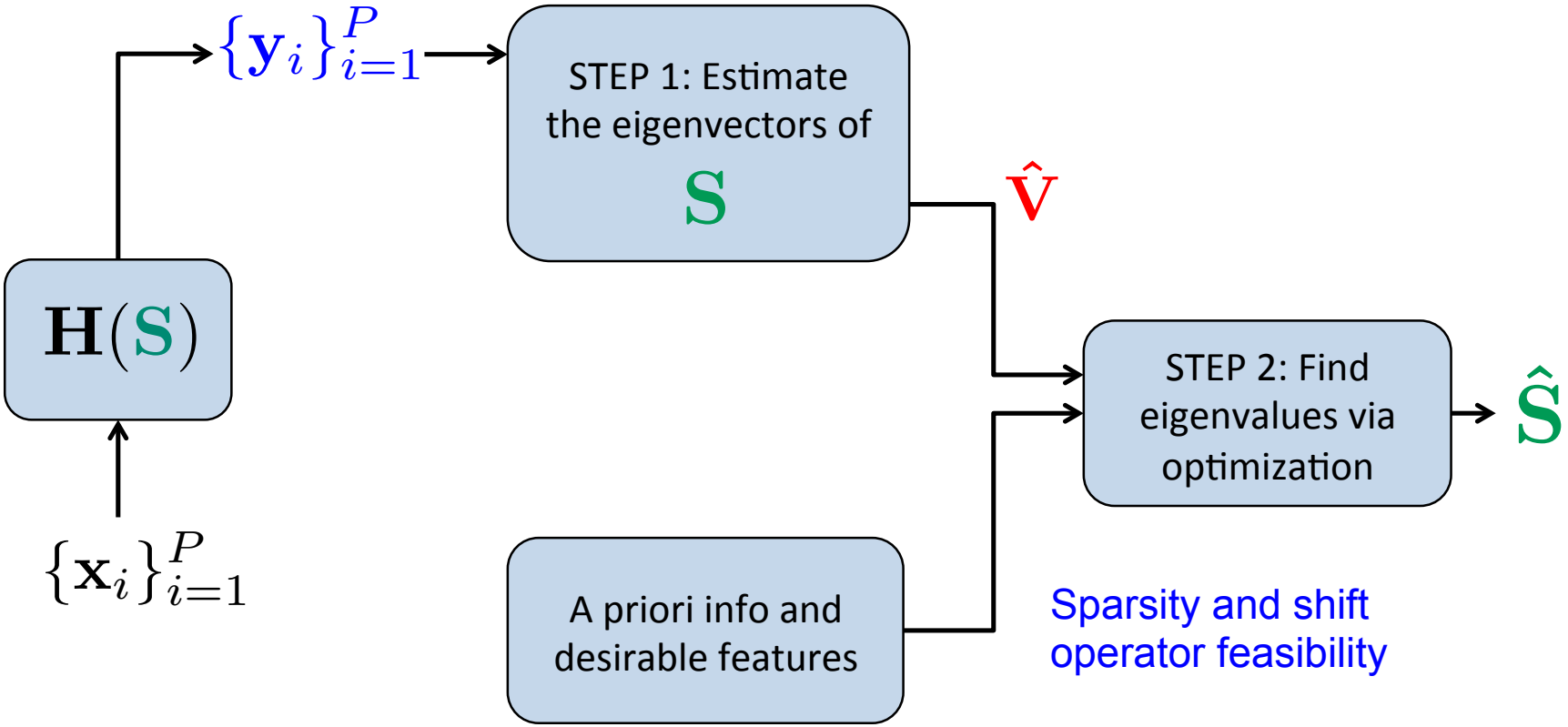
$$(\hat{\mathbf{b}}^*)^T \hat{\mathbf{W}} \hat{\mathbf{b}}^* \leq \mathbb{E}\left[(\tilde{\mathbf{b}}_{l^*})^\top \hat{\mathbf{W}} \tilde{\mathbf{b}}_{l^*}\right] \leq \frac{2}{\pi}(\hat{\mathbf{b}}^*)^T \hat{\mathbf{W}} \hat{\mathbf{b}}^* + \gamma,$$

where $\gamma = \left(1 - \frac{2}{\pi}\right) \lambda_{\max}(\hat{\mathbf{W}}) NM$.

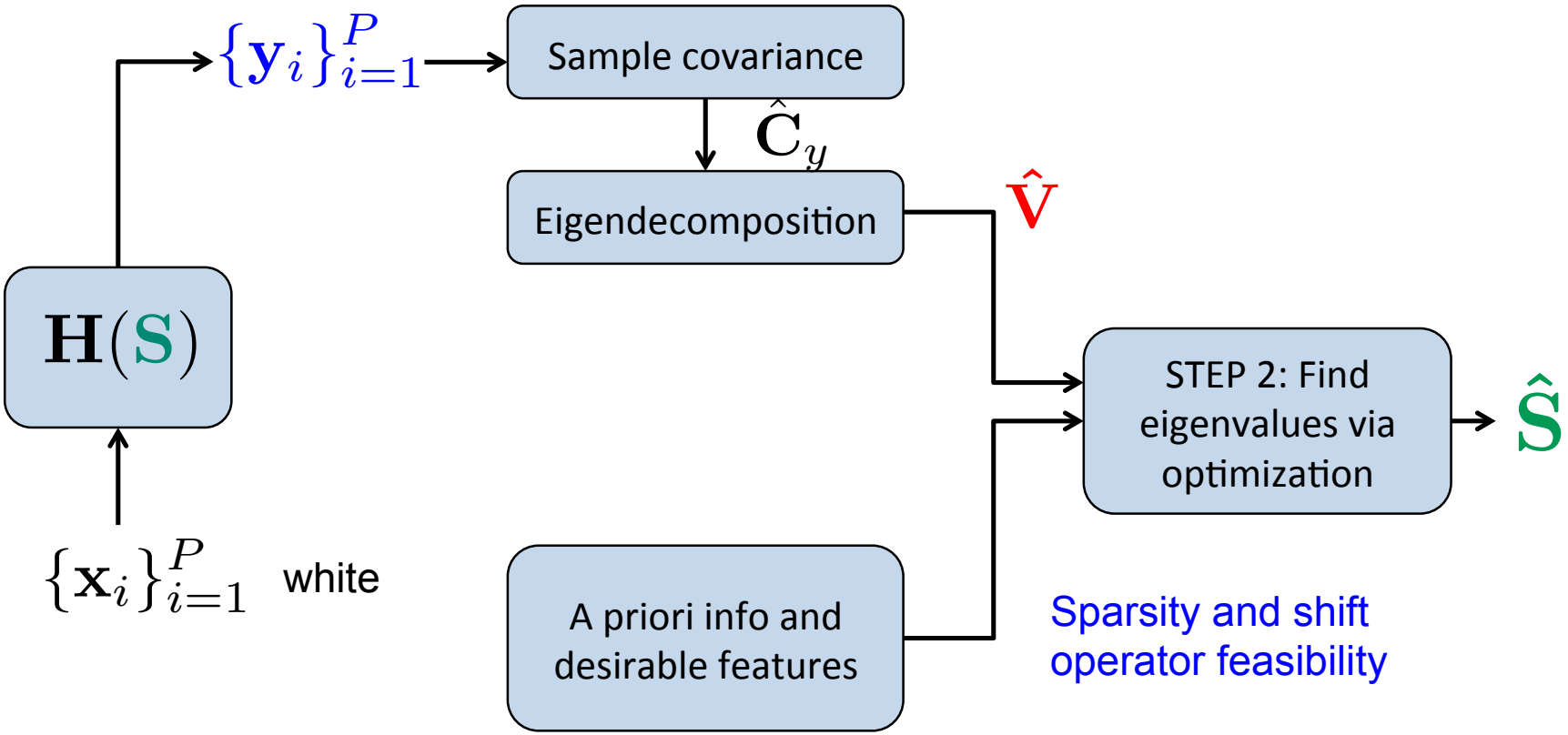
Summary of Step 1



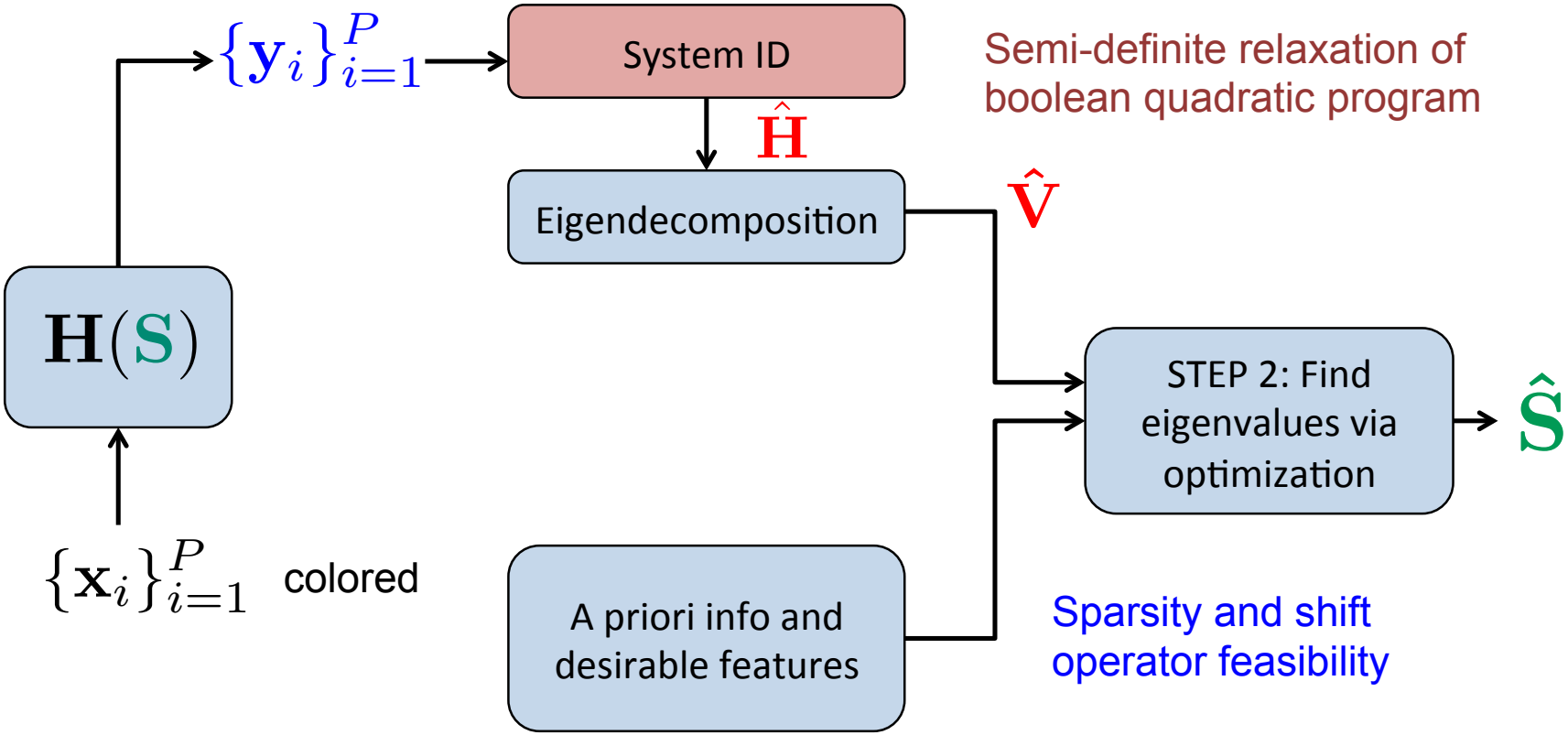
Summary of Step 1



Summary of Step 1



Summary of Step 1



Step 2: Obtaining the eigenvalues

- We can use extra knowledge/assumptions to choose one graph

  $\Rightarrow$ Of all graphs, select one that is optimal in some sense

$$\mathbf{S}^* := \underset{\mathbf{S},\boldsymbol{\lambda}}{\operatorname{argmin}} \; f(\mathbf{S},\boldsymbol{\lambda}) \qquad \text{s. to} \quad \mathbf{S} = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^\top, \quad \mathbf{S} \in \mathcal{S}$$

- Set $\mathcal{S}$ contains all admissible scaled adjacency matrices

$$\mathcal{S} := \{\mathbf{S} \mid S_{ij} \geq 0, \quad \mathbf{S} \in \mathcal{M}^N, \; S_{ii} = 0, \quad \textstyle\sum_j S_{1j} = 1\}$$

  $\Rightarrow$ Can accommodate Laplacian matrices as well

- Problem is convex if we select a convex objective $f(\mathbf{S},\boldsymbol{\lambda})$

  Ex: Sparsity ($f(\mathbf{S}) = \|\mathbf{S}\|_1$), min. energy ($f(\mathbf{S}) = \|\mathbf{S}\|_F$), mixing ($f(\boldsymbol{\lambda}) = -\lambda_2$)

Sparse graph recovery

- Whenever the problem's feasibility set is non-trivial

  $\Rightarrow f(\mathbf{S}, \boldsymbol{\lambda})$ determines the features of the recovered graph

  Ex: Identify sparsest shift $\mathbf{S}_0^*$ that explains observed signal structure

  $\Rightarrow$ Set the objective $f(\mathbf{S}, \boldsymbol{\lambda}) = \|\mathbf{S}\|_0 = |\text{supp}(\mathbf{S})|$

- Non-convex problem, relax to ℓ_1-norm minimization, e.g., [Tropp'06]

$$\mathbf{S}_1^* := \underset{\mathbf{S}, \boldsymbol{\lambda}}{\text{argmin}} \ \|\mathbf{S}\|_1 \quad \text{s. to} \quad \mathbf{S} = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^\top, \ \ \mathbf{S} \in \mathcal{S}$$

- **Q:** Does the solution $\mathbf{S}_1^*$ coincide with the ℓ_0 solution $\mathbf{S}_0^*$?

Recovery guarantee for ℓ_1 relaxation

- $\mathcal{D}$ is the index set such that $\mathrm{vec}(\mathbf{S})_{\mathcal{D}} = \mathrm{diag}(\mathbf{S})$
- $\mathcal{K}$ indexes the support of $\mathbf{s}_0^* = \mathrm{vec}(\mathbf{S}_0^*)$
- Define $\mathbf{M} := \mathbf{V} \odot \mathbf{V}$, where $\odot$ is the Khatri-Rao product

  $\Rightarrow$ Form $\mathbf{R} := [(\mathbf{I} - \mathbf{M}\mathbf{M}^\dagger)_{\mathcal{D}^c},\ \mathbf{e}_1 \otimes \mathbf{1}_{N-1}]$

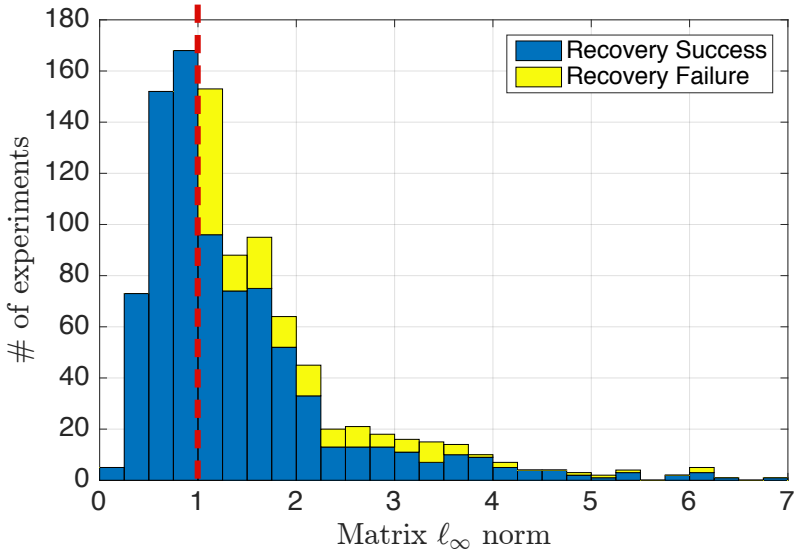
Theorem: $\mathbf{S}_1^* = \mathbf{S}_0^*$ if the two following conditions are satisfied
1) $\mathrm{rank}(\mathbf{R}_{\mathcal{K}}) = |\mathcal{K}|$; and
2) There exists a constant $\delta > 0$ such that

$$\psi_{\mathbf{R}} := \|\mathbf{I}_{\mathcal{K}^c}(\delta^{-2}\mathbf{R}\mathbf{R}^\top + \mathbf{I}_{\mathcal{K}^c}^\top \mathbf{I}_{\mathcal{K}^c})^{-1}\mathbf{I}_{\mathcal{K}}^\top\|_\infty < 1$$

- Cond. 1) ensures uniqueness of solution $\mathbf{S}_1^*$
- Cond. 2) guarantees existence of a dual certificate for ℓ_0 optimality

Sparse recovery guarantee

- Generate 1000 ER random graphs ($N = 20$, $p = 0.1$) such that
  - $\Rightarrow$ Feasible set is not a singleton
  - $\Rightarrow$ Cond. 1) in sparse recovery theorem is satisfied
- Noiseless case: ℓ_1 norm guarantees recovery as long as $\psi_{\mathbf{R}} < 1$



- Condition is sufficient but not necessary
  - $\Rightarrow$ Tightest possible bound on this matrix norm

Noisy spectral templates

- Step 1 actually yields $\hat{\mathbf{V}}$, a noisy version of the spectral templates
  - $\Rightarrow$ With $d(\cdot,\cdot)$ denoting a (convex) distance between matrices

$$\min_{\{\mathbf{S},\boldsymbol{\lambda},\hat{\mathbf{S}}\}} \|\mathbf{S}\|_1 \quad \text{s. to} \quad \hat{\mathbf{S}} = \sum_{k=1}^{N} \lambda_k \hat{\mathbf{v}}_k \hat{\mathbf{v}}_k^\top, \quad \mathbf{S} \in \mathcal{S}, \quad d(\mathbf{S},\hat{\mathbf{S}}) \leq \epsilon$$

- **Q:** How does the noise in $\hat{\mathbf{V}}$ affect the recovery?
- Stable recovery can be established $\Rightarrow$ depends on noise level
  - $\Rightarrow$ Reformulate problem as $\min_{\mathbf{t}} \|\mathbf{t}\|_1$ s. to $\|\hat{\mathbf{R}}^\top \mathbf{t} - \mathbf{b}\|_2 \leq \epsilon$
- Conditions 1) and 2) but based on $\hat{\mathbf{R}}$, guaranteed $d(\mathbf{S}^*, \mathbf{S}_0^*) \leq C\epsilon$
  - $\Rightarrow$ ϵ large enough to guarantee feasibility of $\mathbf{S}_0^*$
  - $\Rightarrow$ Constant C depends on $\hat{\mathbf{V}}$ and the support $\mathcal{K}$

Incomplete spectral templates

- Partial access to $\mathbf{V}$ $\Rightarrow$ Only K known eigenvectors $\mathbf{V}_K = [v_1, \ldots, v_K]$

$$\min_{\{\mathbf{S}, \mathbf{S}_{\bar{K}}, \boldsymbol{\lambda}\}} \|\mathbf{S}\|_1 \text{ s. to } \mathbf{S} = \mathbf{S}_{\bar{K}} + \sum_{k=1}^{K} \lambda_k \mathbf{v}_k \mathbf{v}_k^\top, \ \mathbf{S} \in \mathcal{S}, \ \mathbf{S}_{\bar{K}} \mathbf{V}_K = \mathbf{0}$$

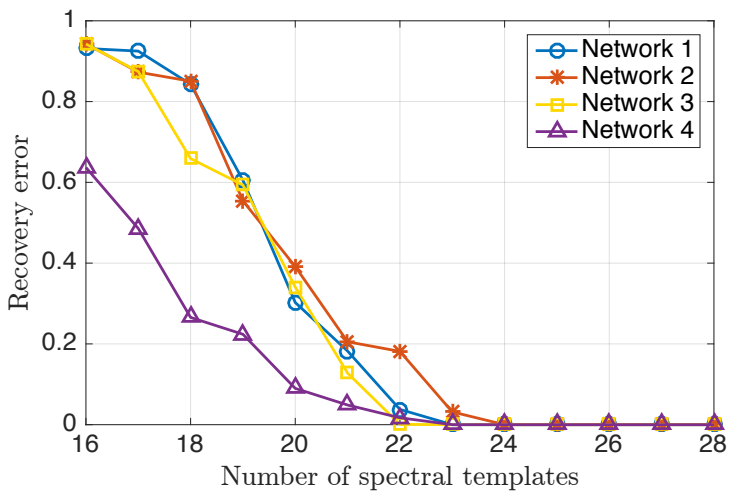
- **Q:** How does the (partial) knowledge of $\mathbf{V}_K$ affect the recovery?
- Define $\mathbf{P} := [\mathbf{P}_1, \ \mathbf{P}_2]$ in terms of $\mathbf{V}_K$, and $\boldsymbol{\Upsilon} := [\mathbf{I}_{N^2}, \mathbf{0}_{N^2 \times N^2}]$
  $\Rightarrow$ Reformulate problem as $\min_{\mathbf{t}} \|\boldsymbol{\Upsilon}\mathbf{t}\|_1$ s.to $\mathbf{P}^\top \mathbf{t} = \mathbf{b}$

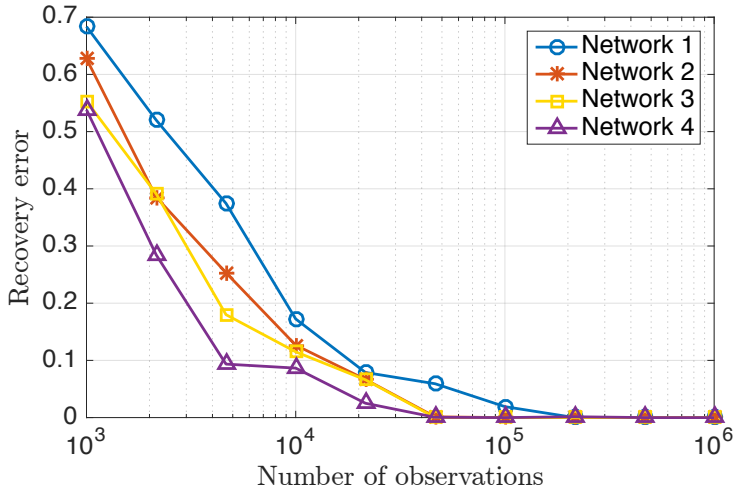
Theorem: $\mathbf{S}^* = \mathbf{S}_0^*$ if the two following conditions are satisfied
1) $\text{rank}([\mathbf{P}_{1\mathcal{K}}^\top, \mathbf{P}_2^\top]) = |\mathcal{K}| + N^2$; and
2) There exists a constant $\delta > 0$ such that

$$\eta_{\mathbf{P}} := \|\boldsymbol{\Upsilon}_{\mathcal{K}^c} (\delta^{-2} \mathbf{P}\mathbf{P}^\top + \boldsymbol{\Upsilon}_{\mathcal{K}^c}^\top \boldsymbol{\Upsilon}_{\mathcal{K}^c})^{-1} \boldsymbol{\Upsilon}_{\mathcal{K}}^\top\|_\infty < 1$$

Social graphs from imperfect templates

- Identification of multiple social networks with $N = 32$
  - $\Rightarrow$ Defined on the same node set of students from Ljubljana
  - $\Rightarrow$ Synthetic signals from diffusion processes in the graphs
- Recovery for incomplete (left) and noisy (right) spectral templates

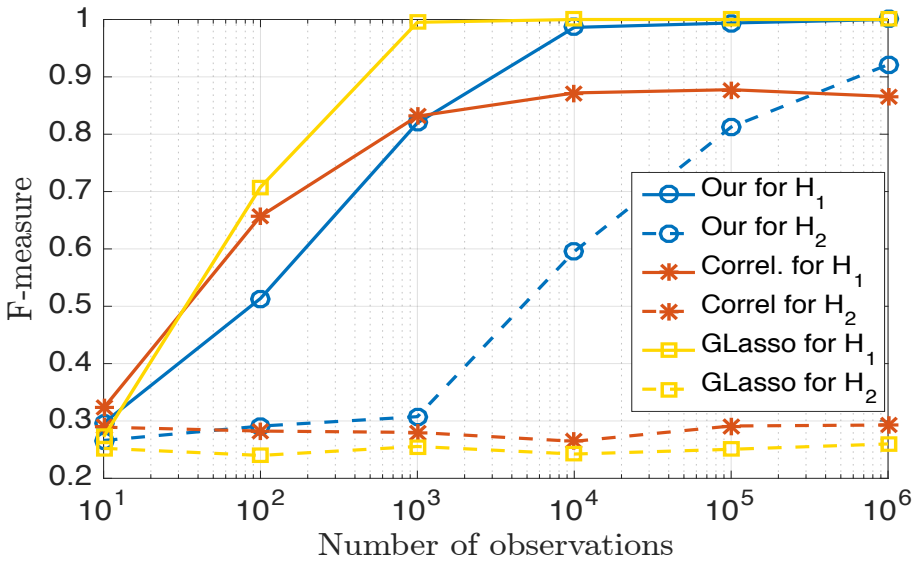




- Error (left) decreases with increasing nr. of spectral templates
- Error (right) decreases with increasing number of observed signals

Performance comparisons

- Comparison with graphical lasso and sparse correlation methods
  - Evaluated on 100 realizations of ER graphs with $N = 20$ and $p = 0.2$



- Graphical lasso implicitly assumes a filter $\mathbf{H}_1 = (\rho \mathbf{I} + \mathbf{S})^{-1/2}$

  $\Rightarrow$ For this filter spectral templates work, but not as well

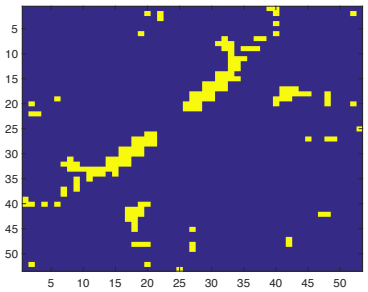
- For general diffusion filters $\mathbf{H}_2$ spectral templates still work fine

Inferring the structure of a protein

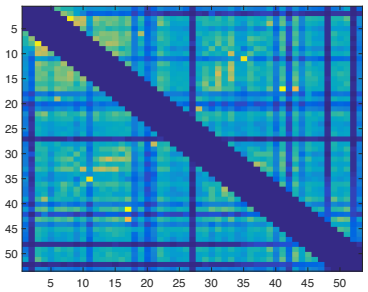
- Our method can be used to sparsify a given network
  - $\Rightarrow$ Keep direct and important edges or relations
  - $\Rightarrow$ Discard indirect relations that can be explained by direct ones
- Use eigenvectors $\hat{\mathbf{V}}$ of given network as noisy eigenvectors of $\mathbf{S}$

Ex: Infer contact between amino-acid residues in BPT1 BOVIN

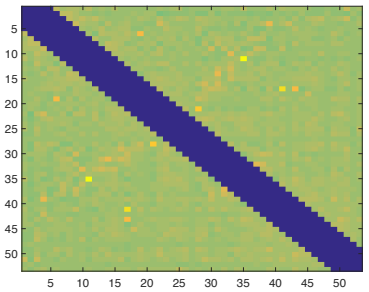
- $\Rightarrow$ Use mutual information of amino-acid covariation as input



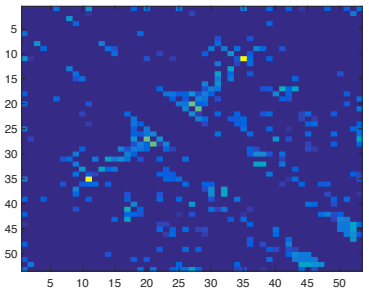
Ground truth



Mutual info.



Network deconv.



Our approach

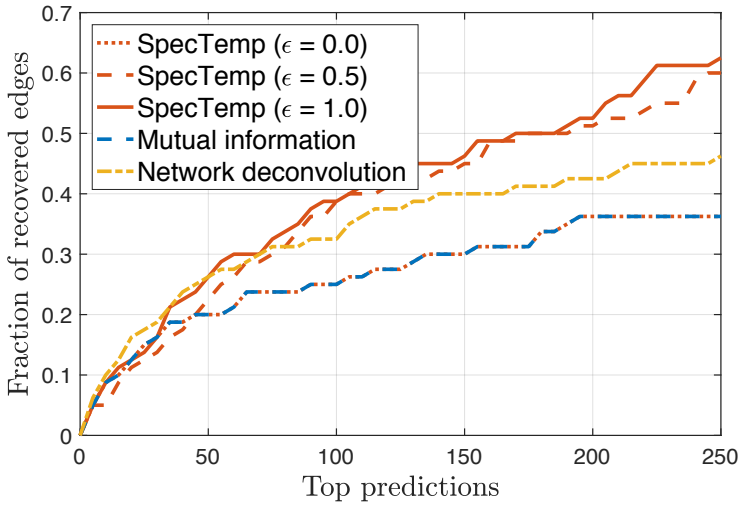
- Network deconvolution assumes a specific filter model [Feizi13]
  - $\Rightarrow$ We achieve better performance by being agnostic to this

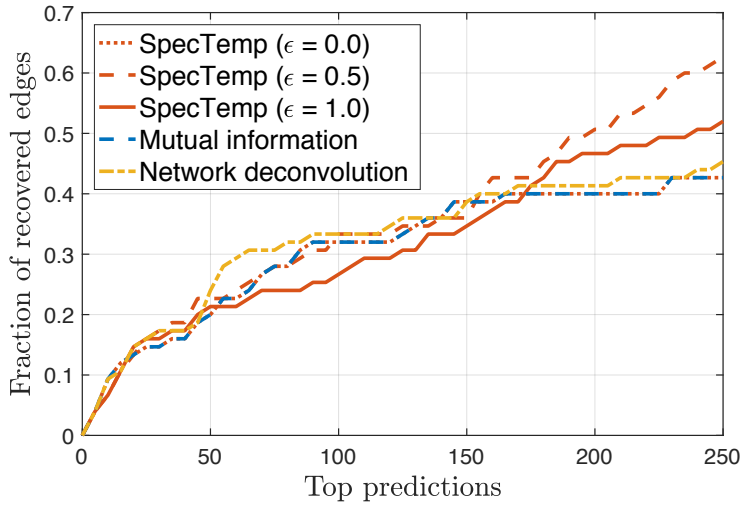
Sensitivity of recovered edges

- Sensitivity of the top edge predictions

  $\Rightarrow$ Fraction of the real contact edges recovered

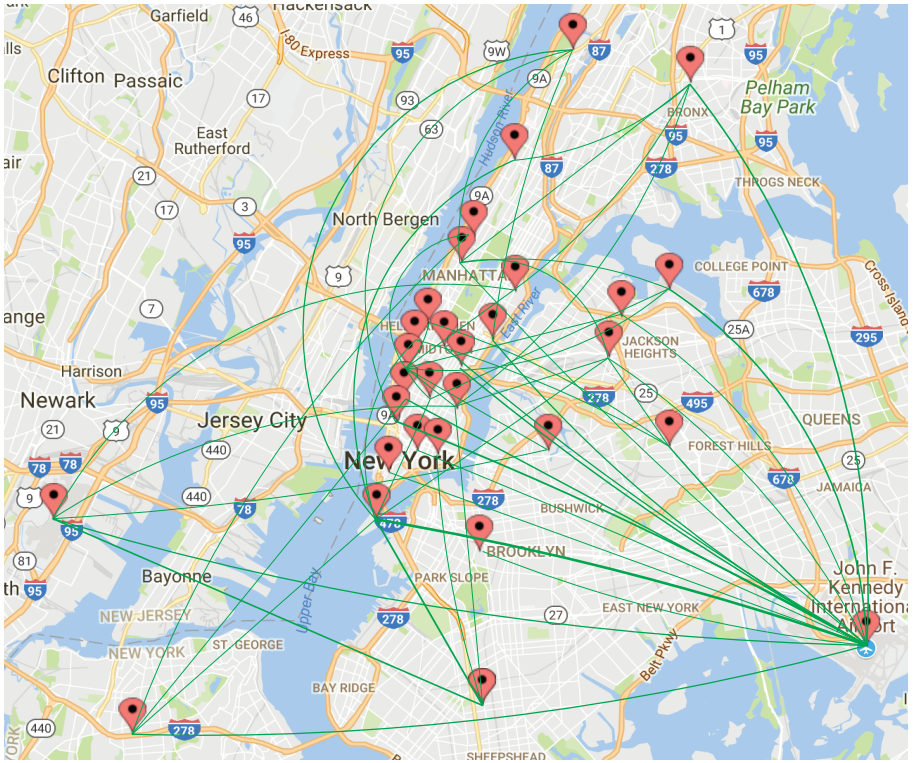
- For $\epsilon = 0$ we force $\mathbf{S}$ to be mutual information matrix $\mathbf{S}'$
- For larger values of ϵ, we get a better recovery





Unveiling urban mobility patterns

- Detect mobility patterns in New York City from Uber pickup data
  - Times and locations ($N = 30$) from January 1st to June 29th 2015
  - Pickups within 6-11am as input signal $\mathbf{x}$ and 3-8pm as output $\mathbf{y}$
  - $M = 2$ graph processes: weekday ($m = 1$) and weekend ($m = 2$) pickups



- Most edges between Manhattan and the other boroughs
- Few edges within Manhattan
  $\Rightarrow$ Uber mostly for commute
- Hubs at JFK, Newark and LaGuardia airports

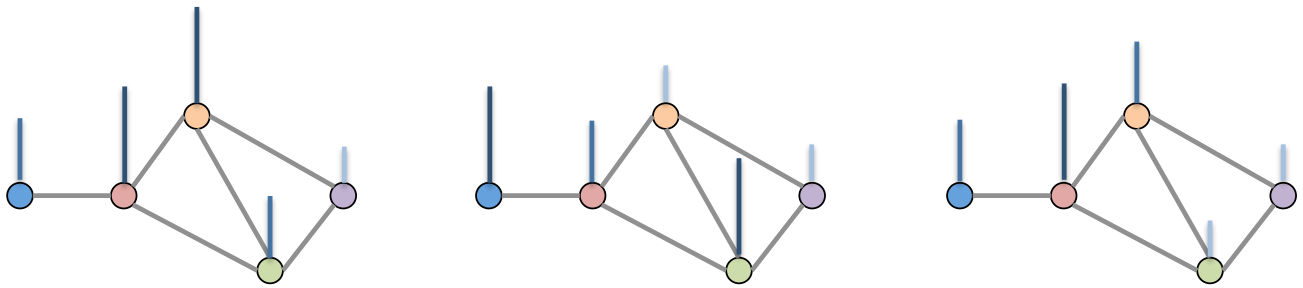
Summary

- GSP approach to network inference in the graph spectral domain
  - $\Rightarrow$ Two step approach: i) Obtain **V**; ii) Estimate **S** given **V**
- How to obtain the spectral templates **V**
  - $\Rightarrow$ Based on covariance of diffused signals
  - $\Rightarrow$ Other sources: network operators, network deconvolution
- Infer **S** via convex optimization
  - $\Rightarrow$ Objectives promote desirable physical properties
  - $\Rightarrow$ Constraints encode a priori information on structure
  - $\Rightarrow$ Robust formulations for noisy and incomplete templates

What about streaming data?

Setup

- Sparse network $\mathcal{G}$ with unknown graph shift $\mathbf{S}$ (even dynamic $\mathbf{S}_t$)
- Observe
  - $\Rightarrow$ Streaming signals $\{\mathbf{y}_t\}_{t=1}^{\top}$ defined on $\mathbf{S}$
  - $\Rightarrow$ Edge status s_{ij} for $(i,j) \in \Omega \subset \mathcal{V} \times \mathcal{V}$



$\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{y}_3$

Problem statement

Given observations $\{\mathbf{y}_t\}_{t=1}^{\top}$ and edge status in Ω, determine the network $\mathbf{S}$ knowing that $\{\mathbf{y}_t\}_{t=1}^{\top}$ are generated via diffusion on $\mathbf{S}$.

Batch topology inference revisited

- Suppose that the input is white, i.e., $\mathbf{C}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right] = \mathbf{I}$

  $\Rightarrow$ The covariance matrix of $\mathbf{y} = \mathbf{H}\mathbf{x}$ is a polynomial in $\mathbf{S}$

$$\mathbf{C}_y = \mathbb{E}\left[\mathbf{H}\mathbf{x}\left(\mathbf{H}\mathbf{x}\right)^\top\right] = \mathbf{H}^2 = h_0^2\mathbf{I} + 2h_0h_1\mathbf{S} + h_1^2\mathbf{S}^2 + \dots$$

- Implies $\mathbf{C}_y\mathbf{S} = \mathbf{S}\mathbf{C}_y$, shift-invariant second-order statistics (stationarity)
- **Formulation:** given $\hat{\mathbf{C}}_y$, search for $\mathbf{S}$ that is sparse and feasible

$$\hat{\mathbf{S}} := \underset{\mathbf{S}}{\operatorname{argmin}} \ \|\mathbf{S}\|_1 \qquad \text{subject to:} \quad \|\mathbf{S}\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y\mathbf{S}\|_F \leq \epsilon, \ \ \mathbf{S} \in \mathcal{S}$$

- Set $\mathcal{S}$ contains all admissible adjacency matrices

$$\mathcal{S} := \{\mathbf{S} \mid S_{ij} \geq 0, \mathbf{S}^\top = \mathbf{S}, S_{ii} = 0, S_{ij} = s_{ij}, \ (i,j) \in \Omega\}$$

Batch proximal gradient algorithm

- Dualize the constraint to arrive at the convex, composite cost $F(\mathbf{S})$

$$\mathbf{S}^\star \in \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmin}}\, F(\mathbf{S}) := \|\mathbf{S}\|_1 + \underbrace{\frac{\mu}{2}\|\mathbf{S}\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y\mathbf{S}\|_F^2}_{g(\mathbf{S})}$$

- Smooth component $g(\mathbf{S})$ has an $M = 4\mu\lambda_{\max}^2(\hat{\mathbf{C}}_y)$-Lipschitz gradient

$$\nabla g(\mathbf{S}) = \mu\left[(\mathbf{S}\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y\mathbf{S})\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y(\mathbf{S}\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y\mathbf{S})\right]$$

- Convergent PG updates with stepsize $\gamma < \frac{2}{M}$ at iteration $k = 1, 2, \ldots$

$$\mathbf{S}_{k+1} = \operatorname{prox}_{\gamma\|\cdot\|_1,\mathcal{S}}\left(\mathbf{S}_k - \gamma\nabla g(\mathbf{S}_k)\right)$$

- Proximal operator $(\mathbf{D}_k := \mathbf{S}_k - \gamma\nabla g(\mathbf{S}_k))$

$$[\mathbf{S}_{k+1}]_{ij} = \begin{cases} 0, & i = j \\ s_{ij}, & (i,j) \in \Omega \\ \max(0, [\mathbf{D}_k]_{ij} - \gamma), & \text{otherwise.} \end{cases}$$

Online proximal gradient algorithm

- **Q:** Online estimation from streaming data $\mathbf{y}_1, \ldots, \mathbf{y}_t, \mathbf{y}_{t+1}, \ldots$?
  - At time t solve the time-varying composite optimization

$$\mathbf{S}_t^\star \in \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmin}}\, F_t(\mathbf{S}) := \|\mathbf{S}\|_1 + \underbrace{\frac{\mu}{2}\|\mathbf{S}\hat{\mathbf{C}}_{\mathbf{y},t} - \hat{\mathbf{C}}_{\mathbf{y},t}\mathbf{S}\|_F^2}_{g_t(\mathbf{S})}$$

- **Step 1:** Recursively update the sample covariance $\hat{\mathbf{C}}_{y,t}$

$$\hat{\mathbf{C}}_{y,t} = \frac{1}{t}\left((t-1)\hat{\mathbf{C}}_{y,t-1} + \mathbf{y}_t\mathbf{y}_t^T\right)$$

  - Track $\mathbf{S}_t$ $\Rightarrow$ Sliding window or exponentially-weighted moving average

- **Step 2:** Run a single iteration of the PG algorithm [Madden et al'18]

$$\mathbf{S}_{t+1} = \operatorname{prox}_{\gamma_t\|\cdot\|_1,\mathcal{S}}\left(\mathbf{S}_t - \gamma_t\nabla g_t(\mathbf{S}_t)\right)$$

  - Memory footprint and computational complexity does not grow with t

Theorem (Madden et al'18)

Let $\nu_t := \left\| \mathbf{S}^\star_{t+1} - \mathbf{S}^\star_t \right\|_F$ *capture the variability of the optimal solution. If* g_t *is strongly convex with constant* m_t *(details in the paper), then for all* $t \geq 1$ *the iterates* $\mathbf{S}_t$ *generated by the online PG algorithm satisfy*

$$\|\mathbf{S}_t - \mathbf{S}^\star_t\|_F \leq \tilde{L}_{t-1} \left(\|\mathbf{S}_0 - \mathbf{S}^*_0\|_F + \sum_{\tau=0}^{t-1} \frac{\nu_\tau}{\tilde{L}_\tau} \right),$$

where $L_t = \max\left\{|1-\gamma_t m_t|, |1-\gamma_t M_t|\right\}, \tilde{L}_t = \prod_{\tau=0}^{t} L_\tau$.

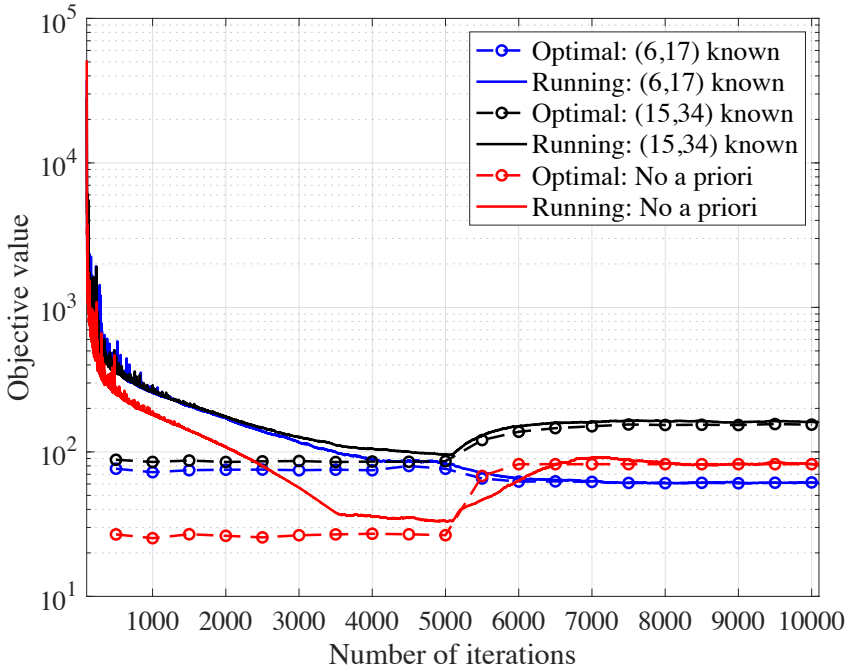
- **Corollary:** Define $\hat{L}_t := \max_{\tau=0,\ldots,t} L_\tau$, $\hat{\nu}_t := \max_{\tau=0,\ldots,t} \nu_\tau$. Then

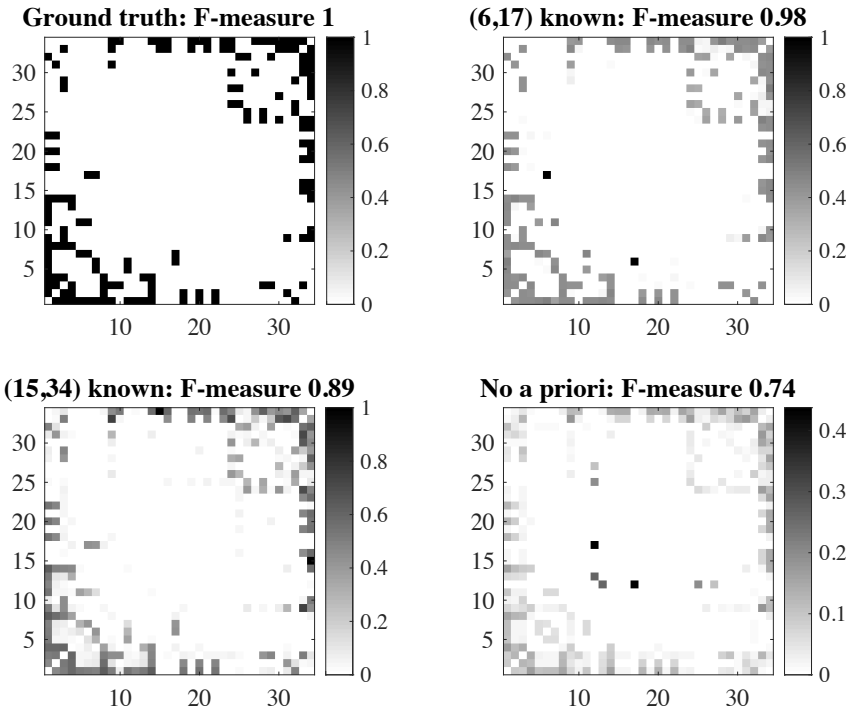
$$\|\mathbf{S}_t - \mathbf{S}^\star_t\|_F \leq \left(\hat{L}_{t-1}\right)^t \|\mathbf{S}_0 - \mathbf{S}^\star_0\|_F + \frac{\hat{\nu}_t}{1-\hat{L}_{t-1}}$$

  - For $m_\tau \geq m$, $M_\tau \leq M$, and $\gamma_\tau = 2/(m_\tau + M_\tau) \Rightarrow \hat{L}_t \leq \frac{M-m}{M+m} < 1$
  - Misadjustment grows with $\hat{\nu}_t$ and bad conditioning ($M \to \infty$ or $m \to 0$)

Zachary's karate club network

- Zachary's karate club social network with $N = 34$ nodes
  - Diffusion filter $\mathbf{H} = \sum_{l=0}^{2} h_l \mathbf{A}^l$, $h_l \sim \mathcal{U}[0,1]$
  - Generate streaming signals $\mathbf{y}_1, \ldots, \mathbf{y}_t, \mathbf{y}_{t+1}, \ldots$ via $\mathbf{y}_t = \mathbf{H}\mathbf{x}_t$
  - Both batch and online inference for different Ω (one edge observed)
  - Dynamic $\mathbf{S}_t$: flip 10% of the edges at random at $t = 5000$





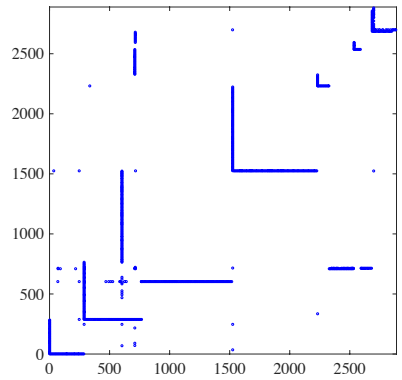
- The online scheme attains the performance of its batch counterpart

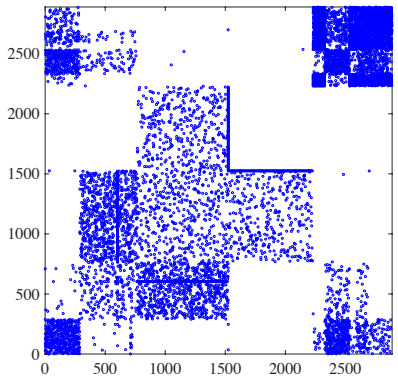
Facebook friendship graph

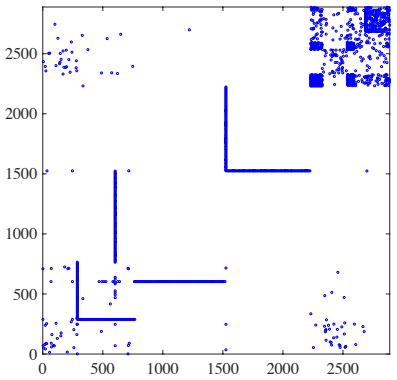
- Facebook friendship graph with $N = 2888$ nodes. Ego-nets of 7 users

Number of observations	10^3	10^4	10^5	10^6
F-measure	0.45	0.77	0.87	0.94

- Ground-truth $\mathbf{A}$ (left) and $\mathbf{S}_t$ for $t = 10^4$ (center) and $t = 10^6$ (right)

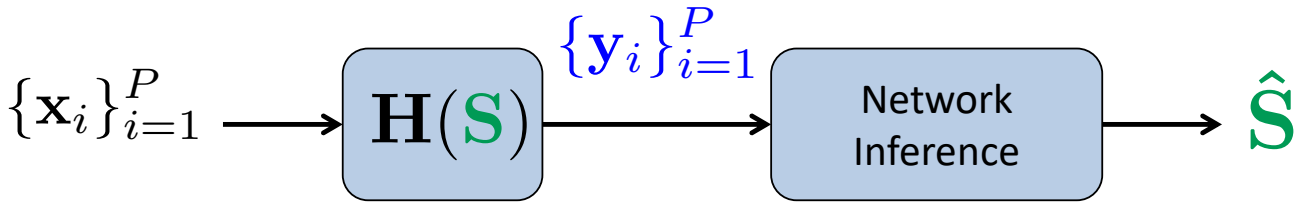




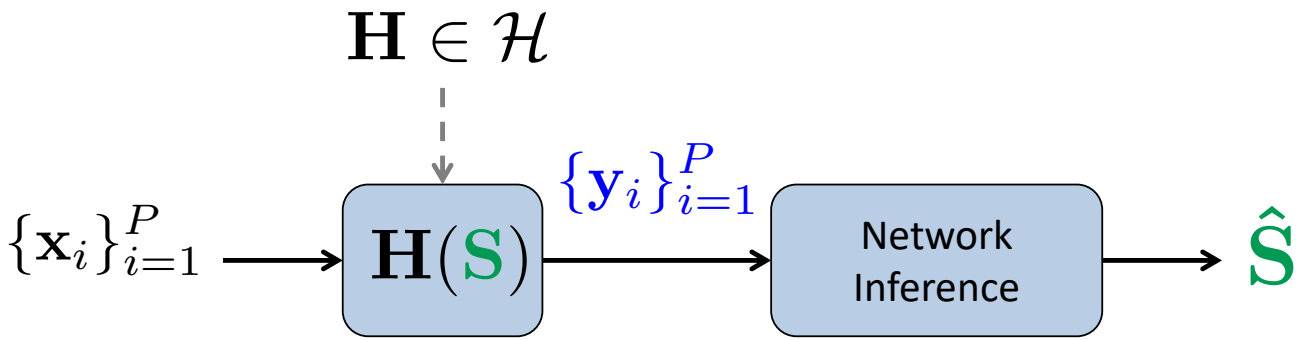


- Scalable to graphs with several thousand nodes

A rich framework for network inference

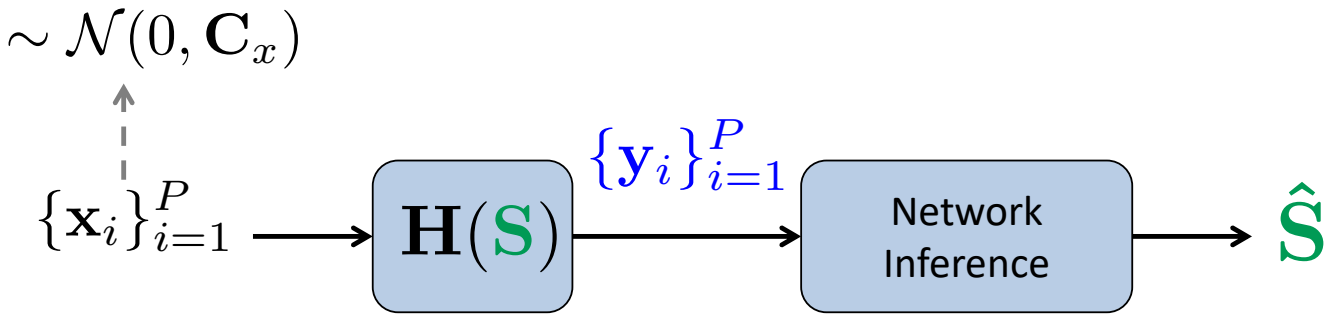


A rich framework for network inference



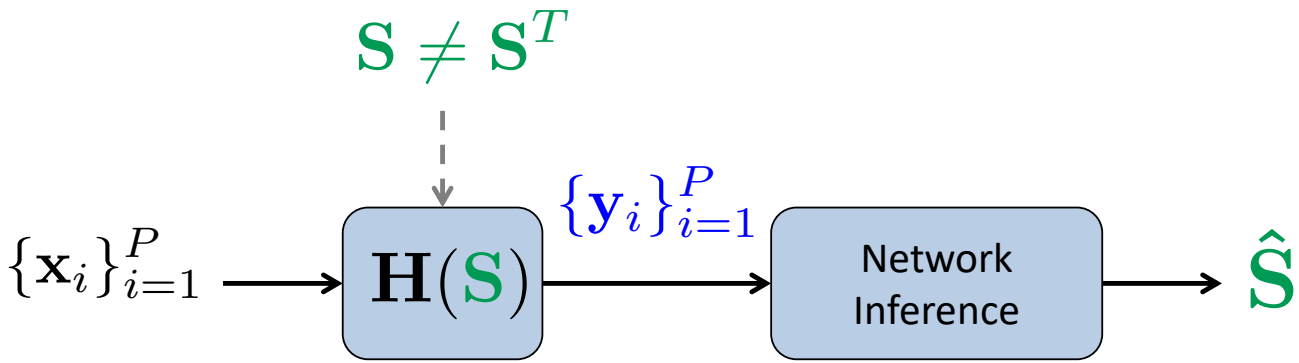
- Prior knowledge on the filter class [Segarra et al'17]

A rich framework for network inference



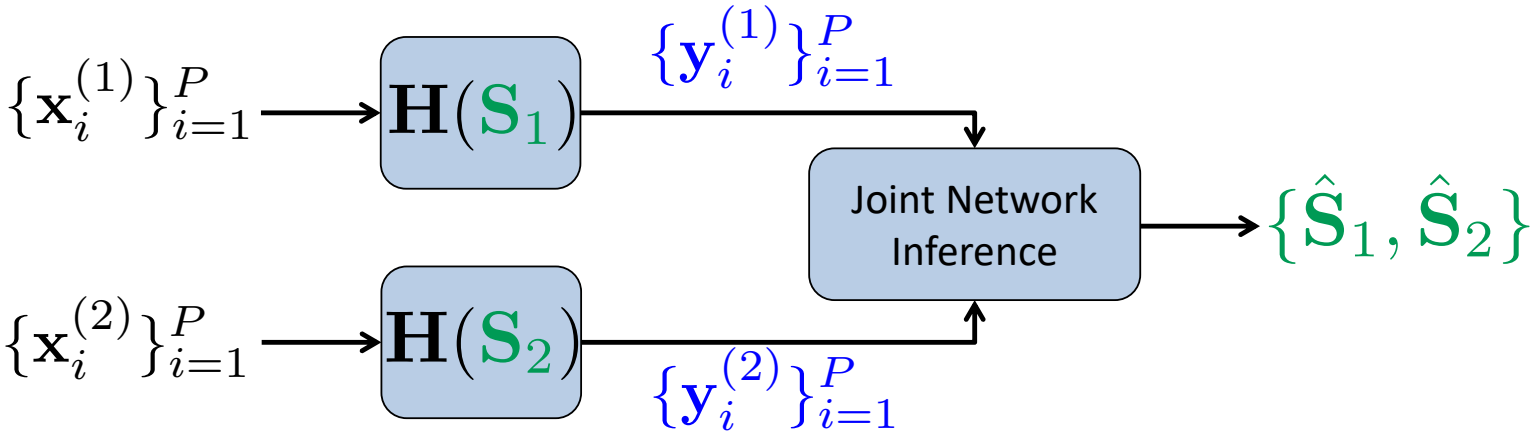
- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]

A rich framework for network inference



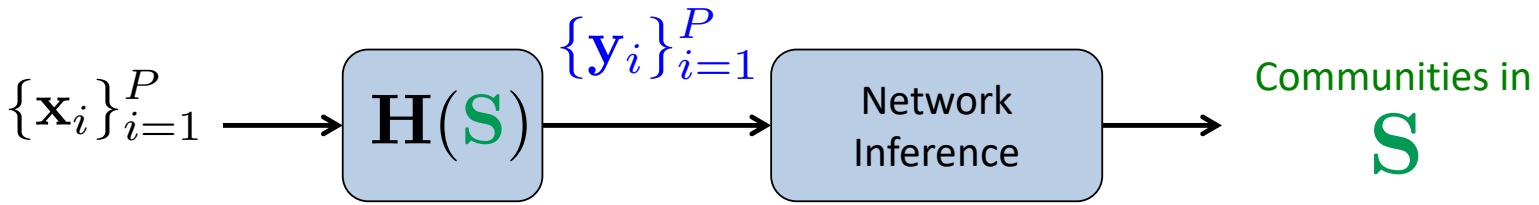
- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]

A rich framework for network inference



- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]
- Joint inference of multiple networks [Segarra et al'17]

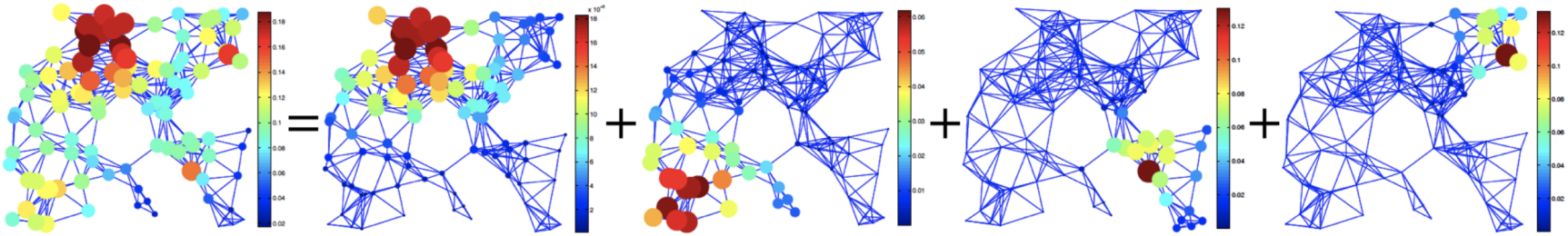
A rich framework for network inference



- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]
- Joint inference of multiple networks [Segarra et al'17]
- Recovering the community structure [Wai et al'18, '19]

Learning heat diffusion graphs

- Superimposed heat diffusion processes on G [Thanou et al'17]



- Dictionary consisting of heat diffusion filters with different rates
  - $\Rightarrow$ Signals modeled as a linear combination of few (sparse) atoms
- Graph learning task as a regularized inverse problem
  - $\Rightarrow$ The graph (hence, the filters) is unknown
  - $\Rightarrow$ The sparse combination coefficients are unknown

Learning heat diffusion graphs: Formulation

- Heat rates $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_S]^\top$ of the S filters $\mathbf{H}_s = e^{-\tau_s \mathbf{L}} = \sum_{l=0}^{\infty} \frac{(-\tau_s \mathbf{L})^l}{l!}$
- Given signals $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$ in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, solve

$$\min_{\mathbf{L}, \mathbf{R}, \boldsymbol{\tau}} \left\{ \left\| \mathbf{X} - \mathbf{K}\mathbf{R} \right\|_F^2 + \alpha \sum_{p=1}^{P} \|\mathbf{r}_p\|_1 + \beta \|\mathbf{L}\|_F^2 \right\}$$

$$\text{s. to} \quad \mathbf{K} = \left[e^{-\tau_1 \mathbf{L}}, e^{-\tau_2 \mathbf{L}}, \ldots, e^{-\tau_S \mathbf{L}} \right]$$

$$\text{trace}(\mathbf{L}) = N, \quad \mathbf{L}\mathbf{1} = \mathbf{0}, \quad L_{ij} = L_{ji} \leq 0, \; i \neq j, \quad \tau_i \geq 0$$

$\Rightarrow \mathbf{R} \in \mathbb{R}^{NS \times P}$ are sparse combination coefficients

$\Rightarrow$ Objective function: Fidelity + sparsity + regularizer

- Non-convex optimization, challenged by matrix exponentials
  - Proximal alternating linearized minimization (PALM)
  - Savings via low-degree polynomial approximation of $\mathbf{H}_s$

Comparative summary

- Main distinctive points of this model
  - $\Rightarrow$ Assumes a specific filter type: heat diffusion
  - $\Rightarrow$ Parametrized by a single scalar: the diffusion rate
  - $\Rightarrow$ Inputs to these filters are required to be sparse
- In comparison, for the spectral templates method
  - $\Rightarrow$ Filters are arbitrary, not just diffusion
  - $\Rightarrow$ Information about inputs is statistical instead of structural
- Inherent trade-off between model and data driven approaches

Concluding remarks

- How to use the information in $\mathcal{X}$ to identify $G(\mathcal{V}, \mathcal{E})$
  - $\Rightarrow$ Mostly focused on static and undirected graphs
  - $\Rightarrow$ GSP offers some novel insights and tools
- Emerging topic areas we did not cover
  - $\Rightarrow$ Directed graphs and causal structure identification
  - $\Rightarrow$ Dynamic networks and multi-layer graphs
  - $\Rightarrow$ Nonlinear models of interaction
- Open research directions
  - $\Rightarrow$ Performance guarantees such as those for graphical lasso
  - $\Rightarrow$ Does smoothness alone suffice? Can sparsity be forgone?
  - $\Rightarrow$ Bi-level network inference: graphs for downstream tasks
  - $\Rightarrow$ Discrete signals, non-linear graph filter based models

Glossary

- Topology inference
- Link prediction
- Association networks
- Network tomography
- Correlation networks
- Multiple testing
- Gene-regulatory network
- Gaussian graphical model
- Covariance selection
- Neigborhood-based regression
- Smoothness prior
- Factor analysis model
- Edge subset selection
- Discriminative graph learning
- Network diffusion process
- Spectral templates
- Graph filter identification
- Semidefinite relaxation
- Robust topology recovery
- Streaming signals
- Online proximal gradient
- Dynamic network
- Heat diffusion graphs
- Directed graphs